WIYN High-Resolution Infrared Camera (WHIRC)

Quick Guide to Data Reduction

Dick Joyce

Version 2.0, 2014 June 26

ACRONYMS AND ABBREVIATIONS: 2
1.0 INTRODUCTION 3
2.0 DATA PREPARATION 3
2.1 INITIAL STEPS 4
2.2 GENERATING DOME FLATS 7
2.3 GENERATING SKY FLATS 9
2.4 ARTIFACT REMOVAL 10
2.4.1 BAD PIXELS 10
2.4.2 PUPIL GHOST 13
2.4.3 SUMMARY OF DOME FLAT ARTIFACT REMOVAL 16
2.4.4 FRINGES 17
3.0 DATA REDUCTION 23
3.1 STANDARDS OR POINTLIKE OBJECTS 23
3.2 EXTENDED OBJECTS OR CROWDED FIELDS 23
3.3 COMBINING IMAGES INTO A MOSAIC 24
3.3.1 DISTORTION CORRECTION 24
3.3.2 SMALL FIELD MOSAICS 26
3.3.3 LARGER MOSAICS 33
4.0 APPENDICES 37
4.1 APPENDIX A – OBSERVATIONAL TEST OF PUPIL GHOST REMOVAL 37

Acronyms and Abbreviations:

DHE	Detector Head Electronics (MONSOON system)
FITS	Flexible Image Transport System (image standard)
GUI	Graphical User Interface
IAS	Instrument Adapter System
IRAF	Image Reduction and Analysis Facility
MOP	MONSOON Observing Platform
PAN	Pixel Acquisition Node (computer), controls MONSOON
OA	Observing Associate
TCS	Telescope Control System
WHIRC	WIYN High Resolution InfraRed Camera
WIYN	Wisconsin Indiana Yale NOAO (Observatory consortium)
WHOCS	WHIRC Observation Control System
WHOMP	WHIRC Observation Manager and Planner
WTTM	WIYN Tip/Tilt Module

WIYN High-Resolution Infrared Camera (WHIRC) Data Reduction Guide

1.0 Introduction

This document is a quick guide to reducing data taken with the WIYN High Resolution Infrared Camera (WHIRC). Data reduction is a highly personal process, so one should keep in mind the procedures described here are biased by the author's personal preferences and experience. Many WHIRC users may have their own (possibly superior) means of reducing data, or use reduction platforms other than IRAF. Suggestions from the user community will be welcomed.

As noted in the User Manual, the high and variable sky (and telescope) background in the infrared results in background flux levels on the detector which are often far larger than the astronomical signals which one is attempting to detect. In addition, the architecture of infrared arrays is one of independently read out pixels, rather than the charge transfer utilized with CCDs. Dead or "hot" pixels (with high dark current) therefore show up as isolated features on infrared images. As with CCDs, observations of a uniformly illuminated target (flatfields) are used to calibrate the pixel sensitivity function of the images.

Both the high sky background and the problem of isolated dead or hot pixels are addressed by taking multiple images of a field, moving the telescope a small distance between integrations. This technique, referred to as "nodding" or "dithering", results in sampling a target at several locations on the array, so the effects of bad pixels can be minimized. Except in very crowded fields or those with extended structure, one is also measuring the sky level on every pixel by moving the sources around after each individual exposure.

2.0 Data Preparation

During a night of observing, one will generally obtain the following types of observations:

1. Flatfields: We recommend a series of dome flats in each of the filters used, at least 10 each with the lights on and an equal number with the lights off. One should use the same Fowler setting in a filter as will be used for the science observations. The latter will serve as the bias and dark subtraction. One can also generate sky flats using sky frames generated from the observations and a series of darks of the same integration time. Sky flats may be more useful for reduction in the H and Ks filters, where there is significant thermal background. One may also use twilight observations for sky flats in the J filter. Nighttime observations in the narrowband filters will generally not accumulate sufficient flux for high S/N sky flats, and even twilight flats in the narrowband filters must be

obtained within a fairly narrow time window. *We strongly suggest that observers who generate sky flats also obtain dome flats as a backup.*

2. Science Observations: These can be a series of dithered observations on the source field (for standards and pointlike sources) or a set of dithered observations on the source and a set of dithered observations at an off-source sky location.

3. Darks: These are not really required for data reduction, because the process of sky subtraction will also subtract out any dark current or bias structure. However, a series of at least 10 short (5 s) darks is a useful diagnostic for evaluating the read noise. *If one generates sky flats, darks at the same integration time are needed.*

2.1 Initial Steps

Independent of the reduction platform or individual techniques used for reduction, there are several steps which should be carried out on all of the data frames prior to any reduction. *Using the wprep.cl script (step 5) will automatically carry out steps 1 – 4, and is recommended.*

1. Trimming: The raw data frames are 2144 × 2050 pixels in size, with 96 columns of reference pixels (Fig. 2.1) on the right side and two rows at the top which are presently of no use and will just make things ugly during the reduction. We suggest trimming these off of all of the data frames; e.g.,

```
imcopy  *.fits[1:2048,1:2048]   *.%fits%tr.fits%
```

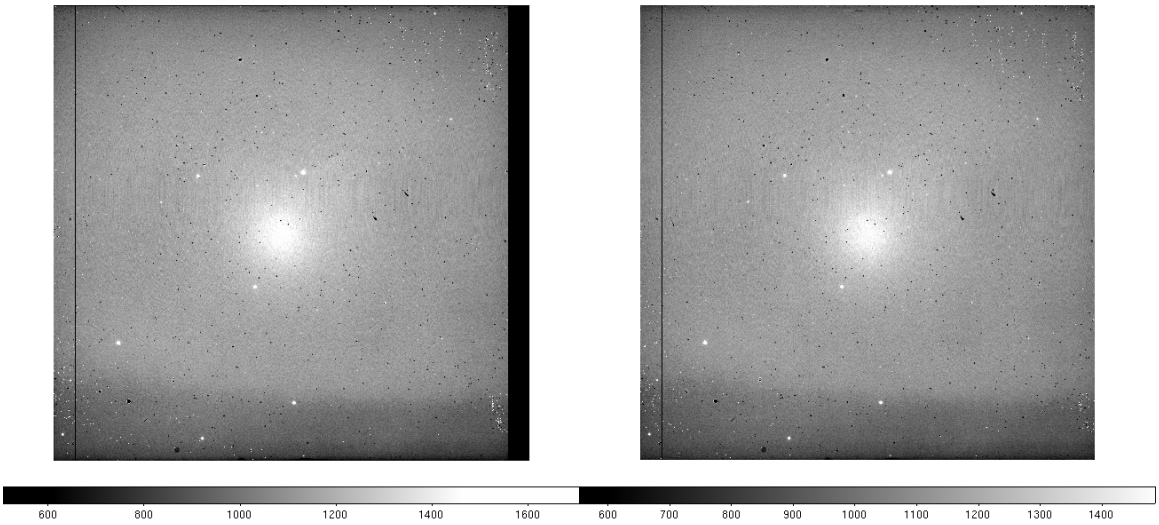

Figure 2.1: Raw Ks band image before (left panel) and after (right panel) trimming. There is one bad column [97] and row [286] on the detector. The bright spot in the center of the field is the pupil ghost.

2. Scaling: If any of the data were taken with Fowler-4 mode, *the pixel values must be normalized by a factor of 4 prior to any linearity correction*. The Fowler-4 mode adds all of the four readouts but does not divide by 4 to avoid possible noise digitization in the integer output format. So for the Fowler-4 data *only:*

```
imarith   *.tr.fits / 4.   *.tr.fits
```

will do this in place.

A safer way to do this is to write *all* of the data to a list, generate a list of the Fowler samples for all of the data, then divide:

```
files *.tr.fits > trlist

hselect @trlist fowler yes > flist

imar @trlist / @flist @trlist
```

3. Linearity correction: As the wells in the array fill up, the gain changes slightly, resulting in a slightly sublinear response, which is approximately 4% at 38000 ADU. We have carried out a quadratic fit to the linearity function for 0.7 v bias up to a value of 38000 ADU. By inverting this function, one can derive a linearity correction function so that the corrected signal S' is related to the raw signal S by

$$S' = S * (A + B*S + C*S^2), \text{ where}$$

$A = 1.0000$	$A' = 1.0000$
$B = 1.29 \times 10^{-7}$	$B' = 0.004227$
$C = 2.506 \times 10^{-11}$	$C' = 0.02691$

The IRAF task irlincor is specifically designed to carry out this correction; the coefficients A′, B′ and C′ above are the irlincor values. *It is critical that linearity correction be performed on the raw data, prior to any sky or dark subtraction, but after any scaling for multiple Fowler sampling.*

Do an epar irlincor and enter the above values for A', B', and C' into the parameter file (control-D to exit), then

```
irlincor *.tr.fits *.%tr%lc%.fits
```

4. At this point you will have three sets of data; you may want to archive the original raw data somewhere, and the trimmed (.tr.fits) data can be deleted; the trimmed/scaled/linearized data (.lc.fits) will be used for reduction.

5. The IRAF script wprep.cl, which can be downloaded from the WHIRC web page, will automatically carry out all of these steps. In addition, it will check the ROTOFF value in the image header and recalculate the WCS parameters so that the sky coordinates are

correctly displayed on the image display. ROTOFF is normally 0.0, but at some telescope orientations, the WIYN instrument rotator has to rotate by 180 degrees because of the limited cable loop. Also, observers may intentionally rotate the instrument to a non-cardinal orientation to fit an elongated target onto the array. These offsets are properly recorded in the ROTOFF keyword, but not in the raw image display at the telescope. After executing wprep.cl, the celestial coordinates will be properly oriented on the image.

5.1 Download the script wprep.cl from the WHIRC webpage and put it in a directory such as the IRAF home directory containing the login.cl file (e.g., /home/user/iraf). But see below.

5.2 Edit the loginuser.cl file to source the task when IRAF is started up:

```
task  wprep = /home/user/iraf/wprep.cl
```

5.3 Put the raw WHIRC images into a list (e.g., files *.fits > inlist). *We strongly recommend that you keep a backup copy of the raw images prior to any reduction, to be safe.*

5.4 Execute the script with wprep. You will be prompted for the input data (@inlist in the present case) and the output data. One may create an output list of the same length as the input list, but with different image names (such as a different extension to the name), in which case one would enter the file name (@outlist). Alternatively, one can enter an extension (such as .wp), and the script will automatically create new output images with the basename extension appended. It is possible to execute the task in place (e.g., using @inlist for both the input and output), but this will lead to confusion with the unprocessed raw data backup, which would have the same image names.

5.5 Images which have been processed with wprep.cl will have a keyword WPREP = 1, 2, or 3 added to the header, so one can verify whether data have been processed.

5.6 One can also quickly verify the process by comparing imstats of the raw and processed data. The processed data will have fewer pixels (4194304 vs 4395200), and the maximum pixel value will be larger because of the linearity correction. In addition, data taken in Fowler-4 mode will be divided by 4.

5.7 **IMPORTANT**: There are now three versions of wprep.cl because of changes since WHIRC was initially installed on WIYN. In June 2009, an adjustment to the rotator resulted in an offset of 0.8º from the initial rotator value. In June 2012, the header values for the rotator offset (ROTOFF) were inexplicably changed from radians to degrees, which is their current state. For data taken prior to June 2009, one should use the script oldwprep1.cl (writes WPREP=1 to the header); for data taken between June 2009 and June 2012, one should use the script oldwprep2.cl (writes WPREP=2); for data taken since June 2012, one should use the script wprep.cl (writes WPREP=3). Hopefully this will remain unchanged in the future.

2.2 Generating Dome Flats

It is generally a good idea to run imstat on the flats just to identify any possible bad frames and eliminate them from the input data.

1. For each of the on/off sets in the filters, imcombine each set into a single image (e.g., flat.k.on, etc.). One can generally use 'combine' = average, with avsigclip rejection, although median should work as well. For the 'lamp on' images, one can scale the images using the mean of a large subregion [500:1500,500:1500] to take out the effects of any drift in the lamp intensity during the observation. The 'lamp off' data generally do not require scaling, since the numbers are small. See Figure 2.3.

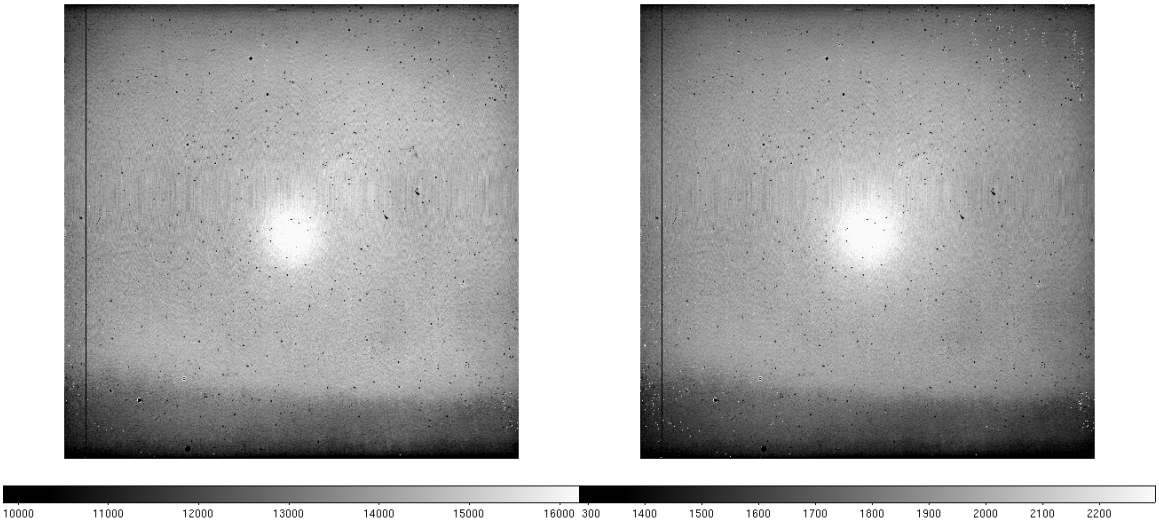

Figure 2.2: Flatfield images at Ks with the flat lamp on (left panel) and off (right panel). Note there is substantial thermal background flux even with the lamp off.

2. Subtract the 'lamp off' from the 'lamp on' image to yield a dark-bias subtracted flat.

3. It is useful to normalize the flats to 1.0 using a fairly large subregion such as [500:1500,500:1500] to keep the science data values from changing too much after the flatfielding operation. Since all data (science and calibrations) are divided by the same flat, the scaling integrity should be preserved.

4. It is also worthwhile to use the IRAF routine imreplace to eliminate very small or negative numbers from the flat so one does not get enormous numbers in the flatfielded science images. I typically use a replacement value of 1.0 and an upper limit of 0.05, keeping the lower level at INDEF.

5. Correct for artifacts if necessary. Since this procedure may apply to both dome flats and sky flats, this is discussed separately in section 2.4.

At the end of this procedure, you will have a normalized flatfield for each of the filters.

joyce@fungo:~/iraf
IRAF
Image Reduction and Analysis Facility
PACKAGE = immatch
TASK = imcombine
input = @flatlist List of images to combine
output = flat.k.on List of output images
(headers=) List of header files (optional)
(bpmasks=) List of bad pixel masks (optional)
(rejmask=) List of rejection masks (optional)
(nrejmas=) List of number rejected masks (optional)
(expmask=) List of exposure masks (optional)
(sigmas =) List of sigma images (optional)
(imcmb =) Keyword for IMCMB keywords
(logfile= STDOUT) Log file
(combine= average) Type of combine operation
(reject = avsigclip) Type of rejection
(project= no) Project highest dimension of input images?
(outtype= real) Output image pixel datatype
(outlimi=) Output limits (x1 x2 y1 y2 ...)
(offsets= none) Input image offsets
(masktyp= none) Mask type
(maskval= 0) Mask value
(blank = 0.) Value if there are no pixels
(scale = median) Image scaling
(zero = none) Image zero point offset
(weight = none) Image weights
(statsec= [500:1500,500:1500]) Image section for computing statistics
(expname=) Image header exposure time keyword
(lthresh= INDEF) Lower threshold
(hthresh= INDEF) Upper threshold
(nlow = 1) minmax: Number of low pixels to reject
(nhigh = 1) minmax: Number of high pixels to reject
(nkeep = 1) Minimum to keep (pos) or maximum to reject (neg)
(mclip = yes) Use median in sigma clipping algorithms?
(lsigma = 3.) Lower sigma clipping factor
(hsigma = 3.) Upper sigma clipping factor
(rdnoise= 40) ccdclip: CCD readout noise (electrons)
(gain = 9) ccdclip: CCD gain (electrons/DN)
(snoise = 0.) ccdclip: Sensitivity noise (fraction)
(sigscal= 0.1) Tolerance for sigma clipping scaling corrections
(pclip = -0.5) pclip: Percentile clipping parameter
(grow = 0.) Radius (pixels) for neighbor rejection
(mode = ql)
ESC-? for HELP

Figure 2.3: IRAF parameter listing for the imcombine task to combine flatfield images (in the list 'flatlist') into a combined 'on' flat. For the 'off' flat images, we recommend setting 'scale' = 'none', since the signal levels are generally small.

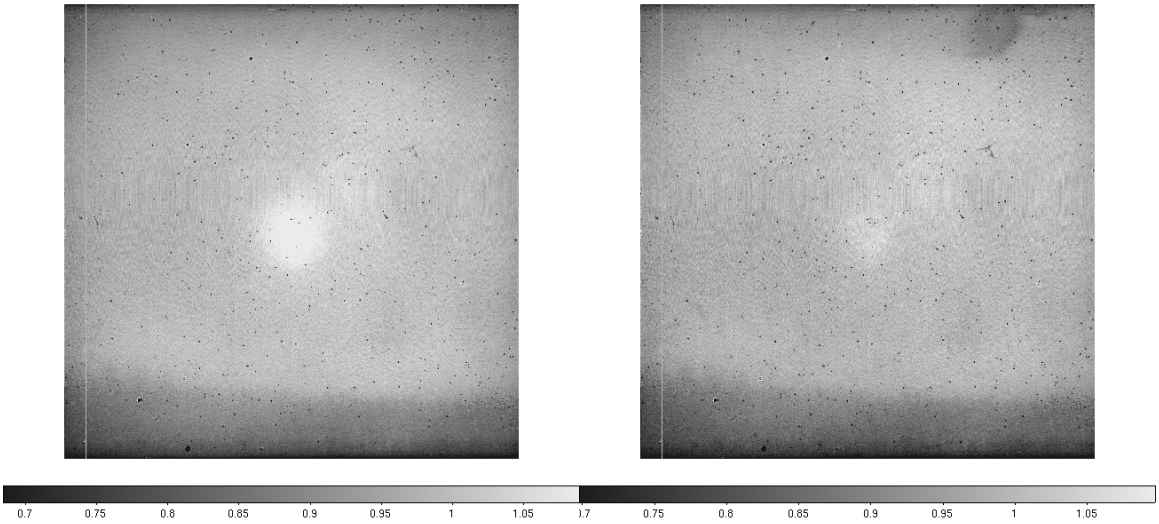

Figure 2.4: Normalized flat for Ks (left panel) and J (right panel). The significant difference in the two emphasizes the need to obtain dedicated flats in each filter used for observing. The pupil ghost in the J band is much less prominent, but there is a noticeable "fingerprint" artifact, probably resulting from a defect in the antireflection coating on the array. Note that the bad column has been set to a value of 1.0 by imreplace.

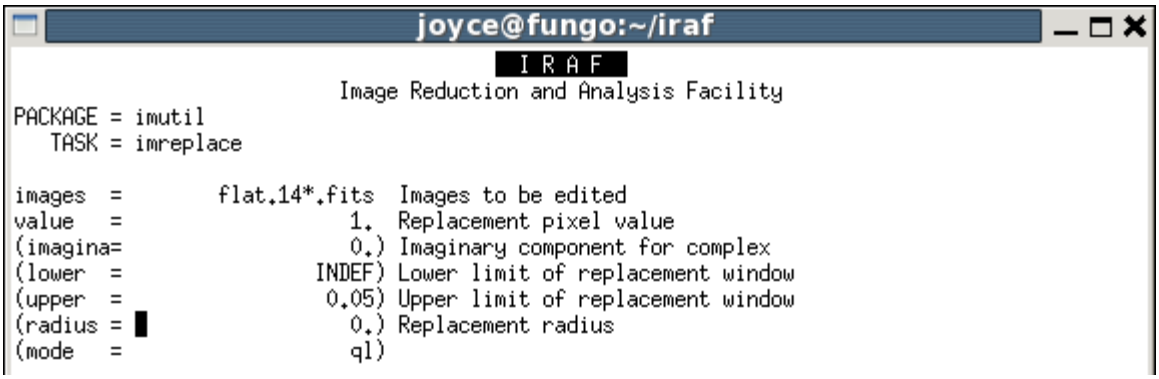
joyce@fungo:~/iraf
IRAF
Image Reduction and Analysis Facility
PACKAGE = imutil
TASK = imreplace
images = flat.14*.fits Images to be edited
value = 1. Replacement pixel value
(imagina= 0.) Imaginary component for complex
(lower = INDEF) Lower limit of replacement window
(upper = 0.05) Upper limit of replacement window
(radius = 0.) Replacement radius
(mode = ql)

Figure 2.5: IRAF parameter listing for the imreplace task which replaces all pixel values < 0.05 by 1.0 to minimize large numbers in the flatfielded image.

2.3 Generating Sky Flats

While sky flats are in theory yield better results than dome flats, since the actual sky illumination is being used, the nighttime sky flux is generally much too small to generate high S/N flats except in the H and Ks filters. We have generated J band twilight flats by taking a series of images once the sky level had fallen below the saturation level and combining them with scaling and a median algorithm, much as is done in the visible. In the Ks band, thermal background from the telescope and WTTM optics will contribute to the background and the pupil ghost, whereas the 'lights on – lights off' procedure for dome flats will subtract out this common signal. Since sky flats can be generated from the target observations, it is safer to get dome flats in any case.

1. Sky flats are best generated from a series of observations which give an appreciable (several thousand ADU) sky level. This pretty much restricts nighttime observations to the H and Ks filters. Twilight flats can be taken in the J and (perhaps) narrowband filters, but the rapid onset of twilight in the infrared leaves limited time for this procedure. H band sky flats may show fringing from the OH lines, which will complicate the process (section 2.4.4).

2. Take a series (~ 10) of dark frames (using the OPAQUE filter) at the same integration time as used for the observations being used to generate the sky flat; this will subtract out any bias and/or dark current. Average the dark frames, perhaps with average sigma clipping to prune out maverick pixel values.

3. Generate a sky image as described in section 3.1. Ideally, one should use a large number of images (10 or more), at more or less the same airmass, with several thousand ADU of background.

4. Subtract the averaged dark frame from the sky frame. Normalize and correct for artifacts as with the dome flats.

2.4 Artifact Removal

WHIRC images contain several types of artifacts, some of which are intrinsic to the detector, others of which result from multiple reflections within the optics. Some correction techniques are discussed in this section, although experienced users may wish to experiment with other procedures. Because infrared arrays utilize a unit-cell architecture, the pixels are effectively independent of each other, unlike those in a CCD.

2.4.1 Bad Pixels

Like all arrays, the detector in WHIRC is not perfect and has several types of cosmetic imperfections. Some of these can be corrected to a degree during the data reduction.

2.4.1.1 High Dark Current Pixels

These are scattered throughout the array, with certain areas, such as the upper right, having a higher concentration. These show up as bright pixels, particularly against a low background such as a dark or narrowband sky. These pixels may also be characterized as "maverick" pixels, since the signal is not necessarily proportional to the frame time, as a true dark current would be. The procedure of sky subtraction does seem to remove most of the elevated signal level, although the noise in these pixels may be higher than in others. Otherwise, there is no remedy for these artifacts.

2.4.1.2 Dimples

A well-illuminated image such as a flat (Fig. 2.4) shows numerous circular artifacts which appear to be dead pixels. However, closer inspection reveals that these are not dead, but areas of decreased response which appear to flatfield out reasonably well (Figs.

2.6, 2.7). The smooth variation evident in a cut through one of the dimples (Fig. 2.7) would make pixel correction algorithms (such as 'fixpix') based on nearest neighbor averaging problematic.

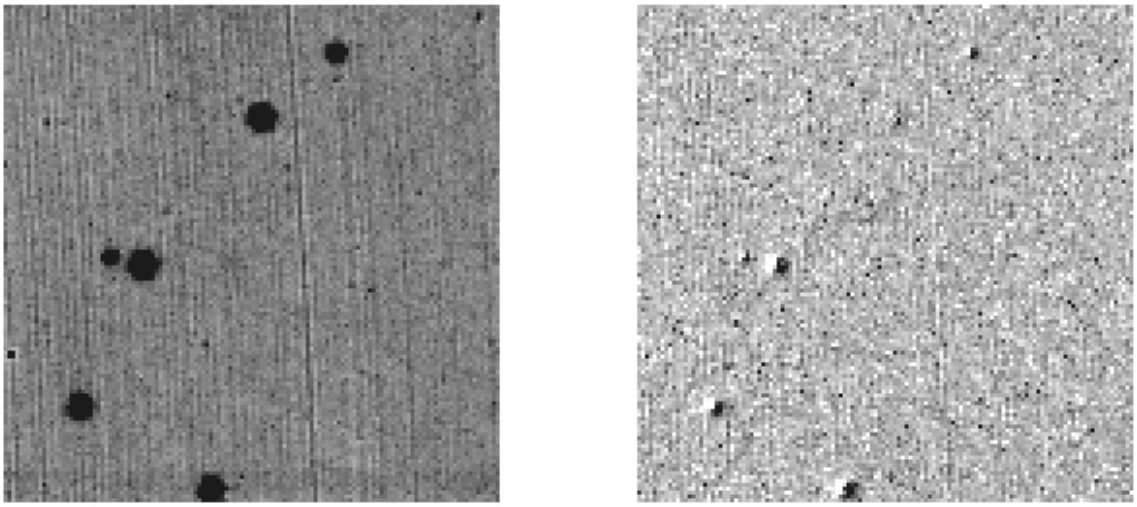

Figure 2.6: Expanded subregion of H band image, showing some of the "dimples" on the detector (left panel). These are not dead pixels and appear to divide out using a flatfield (right panel) to within 1 – 2 %.

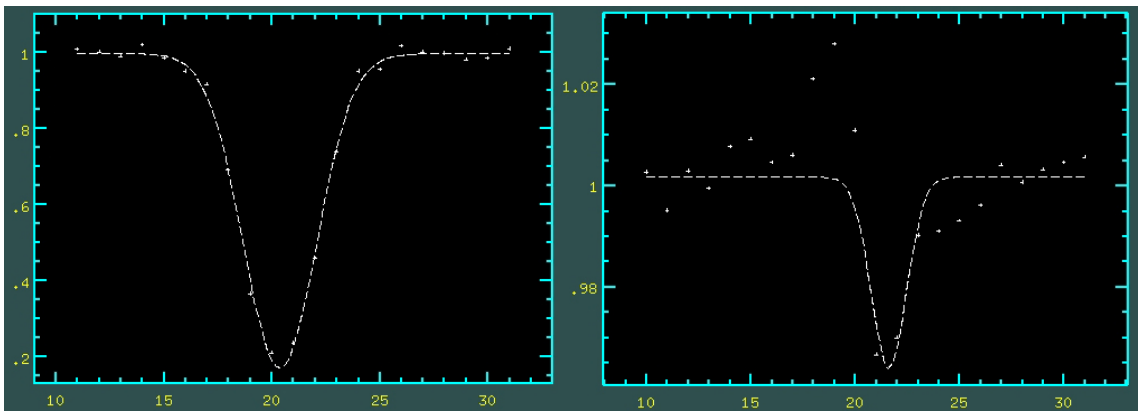

Figure 2.7: Line plot through one of the "dimples" (left panel), illustrating that they are smoothly varying regions of decreased response. The same dimple after flatfielding shows that they divide out to within 1 – 2 % (right panel).

2.4.1.3 Dead Pixels

There are a number of unresponsive pixels in the array, including one row [286] and one column [97] and several clumps which appear to be digs or scratches in the detector surface. There are various techniques for making bad pixel maps described in the IRAF literature. One technique which seems to identify most of the dead pixels is to divide two flatfield images at the same filter and integration times at two different signal levels (either dome flats at different illumination levels or during a twilight sky flat sequence)

and use imreplace to set the pixels deviating from what should be a narrow distribution to the "bad pixel value" in the image. Typically bad pixel maps utilize a value of 1 for bad pixels and 0 for good pixels (Fig. 2.8).

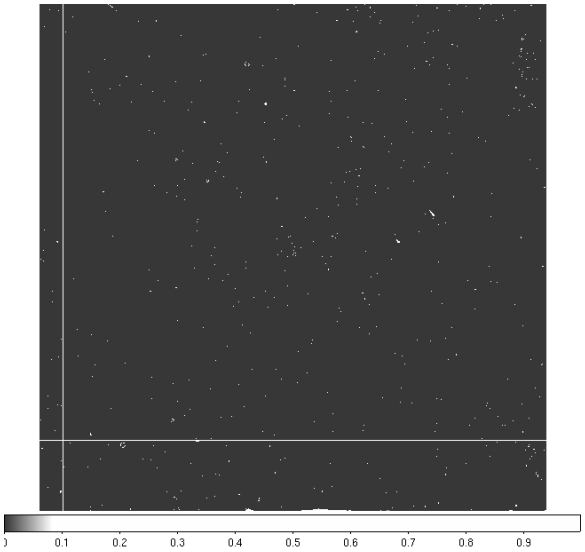

Figure 2.8: Bad pixel image bpix.whirc.fits generated from the ratio of two dome flats.

In addition, one can generate a bad pixel file (which typically has a .pl extension) to identify specific pixels or regions which are bad. Each line in the file identifies the limits of the bad region by the values x1, x2, y1, y2; thus the entry 326 349 255 278 identifies the region between columns 326 and 349 and rows 255 and 278 as bad. This technique is useful for identifying regions, such as PEDs (section 2.4.1.4) which produce some signal and may not be completely identified through the flatfield ratio technique. IRAF reduction tasks such as fixpix (Fig 2.9) will accept either a bad pixel image or file (or both). WARNING: fixpix works on images in place, so one should copy the images to be modified first: imcopy flat.140420.*.fits flat.140420.*.%fits%fx.fits%. Figure 2.10 shows a flatfield image after removal of the bad pixels and the pupil ghost (*vide infra*).

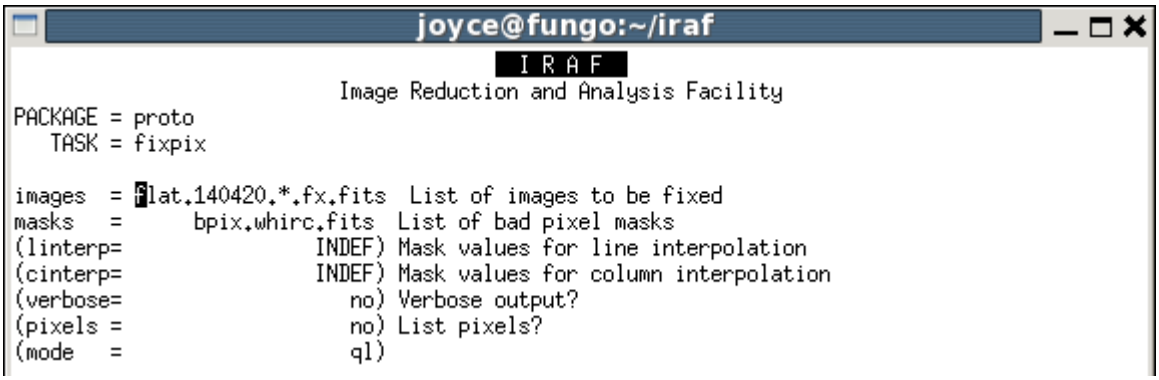
joyce@fungo:~/iraf
IRAF
Image Reduction and Analysis Facility
PACKAGE = proto
TASK = fixpix
images = flat.140420.*.fx.fits List of images to be fixed
masks = bpix.whirc.fits List of bad pixel masks
(linterp= INDEF) Mask values for line interpolation
(cinterp= INDEF) Mask values for column interpolation
(verbose= no) Verbose output?
(pixels = no) List pixels?
(mode = ql)

Figure 2.9: IRAF parameter listing for the fixpix task, which replaces bad pixels in images as defined by the bad pixel map bpix.whirc.fits. This task operates on images in place.

2.4.1.4 Photo-emitting Defects

Photo-emitting defects (PEDs) are generally pixels which become shorted during the hybridization process. They draw significant current during the bias reset and emit light. PEDs are generally identified during the testing stage by the vendor and "cauterized" using a laser. This produces a region of limited sensitivity approximately 20 pixels in diameter, surrounded by a bright annulus. The WHIRC detector has at least two PEDs, centered near [337:266] and [839:1804]. The first of these appears to still emit some light and may be responsible for the region of elevated dark current known as the "palmprint".

Because these regions are not completely dead, the technique of generating a bad pixel image by the flatfield ratio technique does not seem to encompass the entire PED, so using a bad pixel file to identify these regions is good insurance.

2.4.2 Pupil Ghost

As noted in section 2.2, WHIRC images feature a bright region near the center of the detector. This pupil ghost is a common feature of on-axis refractive optical imagers and is a result of multiple internal reflections within the optical train. Unlike the MOSAIC imager pupil ghost, which arises within the corrector and appears as a highly defocused image of the telescope exit pupil, the WHIRC ghost probably arises within both the collimator and camera optics and shows up as a broad central peak (Fig 2.4).

Because the pupil ghost arises from the total background flux, it is seen in both sky and dome flat images. Under reasonably constant sky background conditions, the technique of sky subtraction will effectively remove the pupil ghost from the science data. However, the 'lights on – lights off' technique for producing dome flats will subtract off dark current and any common-mode signal, such as thermal emission from the telescope and WTTM optics in the K band, but not the pupil ghost from the flatfield lights. As a result, the pupil ghost appears in the reduced flat. Because, this does not represent a true response peak, using a flatfield with the pupil ghost will result in an artificially reduced signal level in the central region of the detector. It is therefore necessary to remove this artifact for accurate calibration of science data.

The IRAF package 'mscred', designed for reducing MOSAIC data, has routines for modeling the pupil ghost and removing it from the data. Unfortunately, the modeling task works only for ghosts with a central obscuration, as seen with MOSAIC, and is not effective for the centrally peaked WHIRC ghost. Applying a smoothing function using a task such as 'fmedian' appears to be a reasonable approach, but a careful inspection of the flats in Fig. 2.4 shows a bright region in the vicinity of [1250:1250] which appears in both the J and Ks flats at the same apparent level, and probably represents a true response variation which must remain in the final flat. Since the pupil ghost is stronger at Ks than at J, the ratio of the reduced flatfields after bad pixel correction (Fig. 2.11a) may be an accurate template for the ghost (the J-band "fingerprint" lies well outside the region of interest). Isolating the central 800 × 800 pixel region and smoothing with fmedian (Fig. 2.10) using a 21 × 21 pixel box produces a promising pupil image template (Fig. 2.11b).

Using gauss with sigma=31 gave a similar result; observers may want to try both of these strategies.

```
joyce@fungo:~/iraf
PACKAGE = imfilter
   TASK = fmedian

input   =     flatrat.ksj.edit  Input images to be filtered
output  =      flatrat.ksj.med  Output images
xwindow =                   21  X window size of median filter
ywindow =                   21  Y window size of median filter
(hmin   =               -32768) Minimum histogram bin
(hmax   =                32767) Maximum histogram bin
(zmin   =                INDEF) Pixel value corresponding to hmin
(zmax   =                INDEF) Pixel value corresponding to hmax
(zloreje=                INDEF) Lowside pixel value cutoff
(zhireje=                INDEF) High side pixel value cutoff
(unmap  =                  yes) Unmap the digitized values ?
(boundar=              nearest) Boundary (constant,nearest,reflect,wrap)
(constan=                   0.) Constant for boundary extension
(verbose=                  yes) Print messages about actions taken by the task
(mode   =                   ql)
```

Figure 2.10 : IRAF parameter listing for the fmedian task, to smooth out the ratio of the Ks/J flats. The ratio had been previously edited with imreplace to knock down some of the bright spots in the ratio.

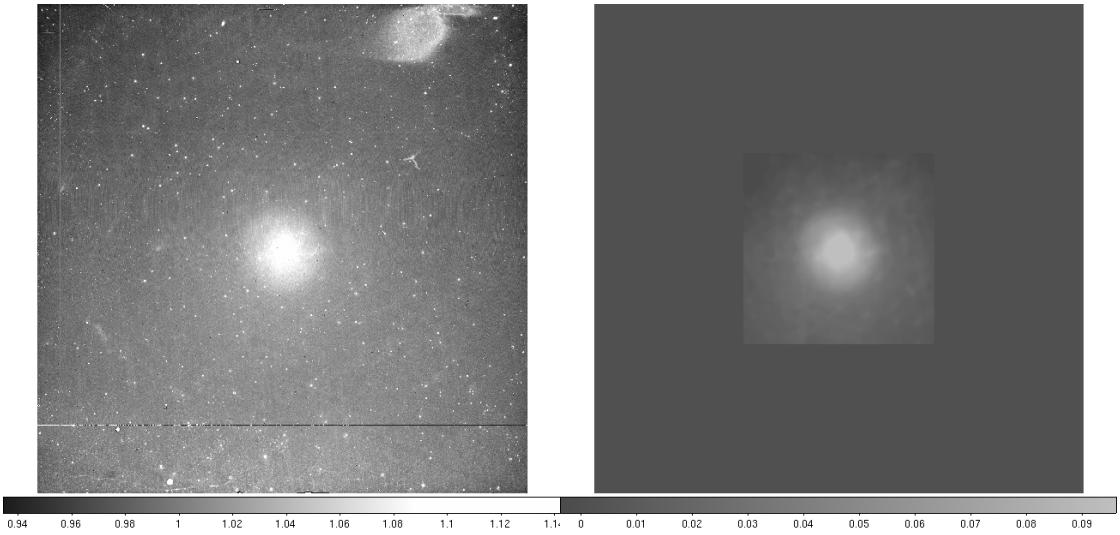

Fig 2.11: (left panel) Ratio of reduced dome flats in Ks and J. Note that the bright region near [1250:1250] cancels out of the ratio, suggesting it is a response variation common to both wavelengths. (right panel) The central 800 × 800 pixel region of the ratio, smoothed to 21 pixels using fmedian.

One procedure which seemed to work was to generate an image with zero values using the task mkimage, in the obscure noao.imred.ccdred.ccdtest package (Fig. 2.12). After editing out the bright spots in the Ks/J flat ratio using imreplace (I set the lower limit to 1.2, the upper limit to INDEF) and running fmedian (Fig. 2.10), run imstat on the resultant image (flatrat.ksj.med) and subtract the median to give a pupil image which

tapers to near zero at the edges. One can then copy the central 800 × 800 region into the zero image to get something like the image in Fig. 2.11b. You may need to play with the level of the zero image to get a smooth transition to the pupil subregion.

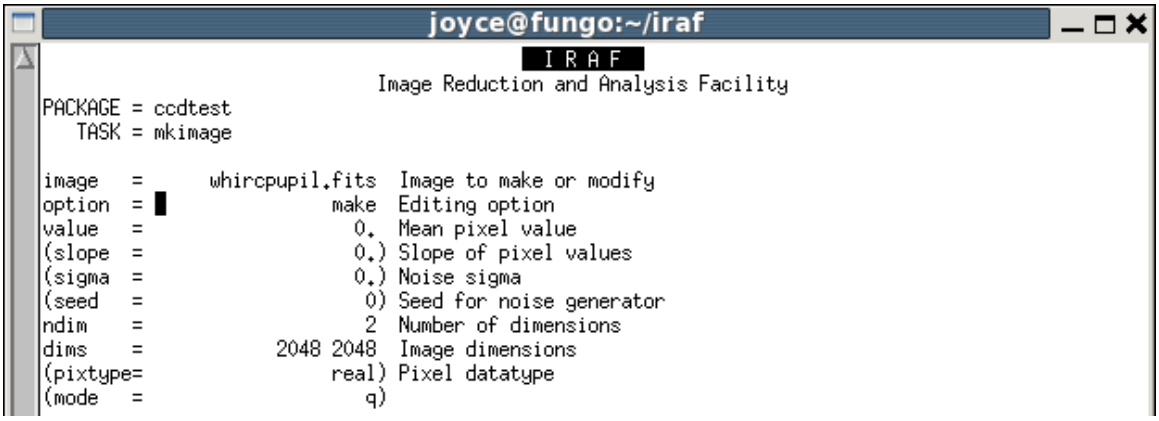
joyce@fungo:~/iraf
IRAF
Image Reduction and Analysis Facility
PACKAGE = ccdtest
TASK = mkimage
image = whircpupil.fits Image to make or modify
option = make Editing option
value = 0. Mean pixel value
(slope = 0.) Slope of pixel values
(sigma = 0.) Noise sigma
(seed = 0) Seed for noise generator
ndim = 2 Number of dimensions
dims = 2048 2048 Image dimensions
(pixtype= real) Pixel datatype
(mode = q)

Figure 2.12: IRAF parameter listing for the task mkimage. This will generate a 2048 × 2048 image 'whircpupil.fits' with pixel values of 0.

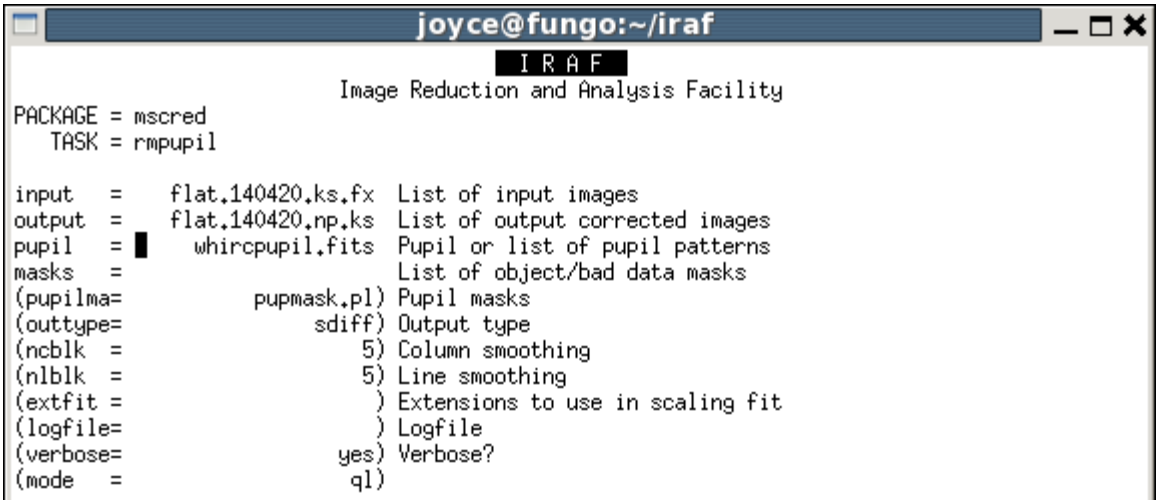
joyce@fungo:~/iraf
IRAF
Image Reduction and Analysis Facility
PACKAGE = mscred
TASK = rmpupil
input = flat.140420.ks.fx List of input images
output = flat.140420.np.ks List of output corrected images
pupil = whircpupil.fits Pupil or list of pupil patterns
masks = List of object/bad data masks
(pupilma= pupmask.pl) Pupil masks
(outtype= sdiff) Output type
(ncblk = 5) Column smoothing
(nlblk = 5) Line smoothing
(extfit =) Extensions to use in scaling fit
(logfile=) Logfile
(verbose= yes) Verbose?
(mode = ql)

Figure 2.13: IRAF parameter listing for the task rmpupil.

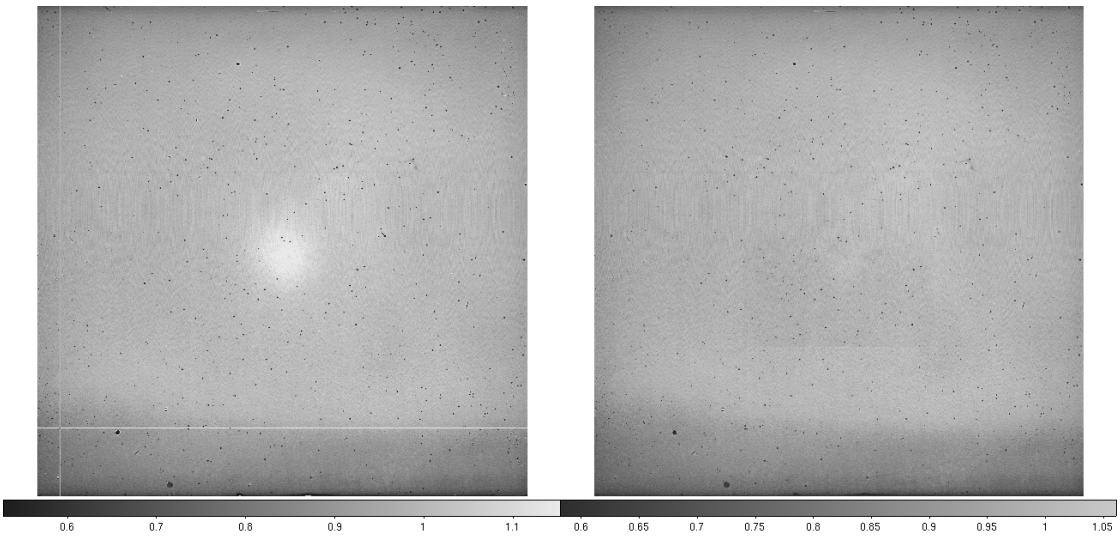

Fig. 2.14: Dome flat for Ks before (left panel) and after (right panel) removal of bad pixels and the pupil ghost. Note the bright region near [1250:1250] remains.

The pupil template can then be used with the task 'mscred.rmpupil' (Fig. 2.13) to remove the pupil ghost from the flat. A representative pupil template (whircpupil.fits) can be downloaded from the WHIRC webpage, although observers may wish to create their own. The pupil mask (pupmask.pl, also on the webpage) is a four-line file which restricts the rmpupil task to the central 600 × 600 pixel region. The 'outtype' parameter should be set to 'sdiff' since one is subtracting the ghost, and the 5 pixel smoothing for row and columns seems to be good. The 'extfit' parameter should be set to "" (null) because the WHIRC images are not multi-extension FITS. The result is shown in Fig. 2.14b.

Observational tests appear to verify this strategy to be effective in removing the effects of the pupil ghost (Appendix A).

Since the shape of the pupil ghost seems to be independent of the filter to first order, this procedure should work for all filters using the same pupil template. The rmpupil task fits the amplitude of the pupil template to the actual amplitude of the pupil ghost automatically.

2.4.3 Summary of Dome Flat Artifact Removal

1. Combine "lamp on" and "lamp off" flat images for each filter using imcombine (Fig. 2.3). "Lamp off" images should not be scaled.

2. Subtract the averaged "off" image from the averaged "on" image:
   imar flat.k.on – flat.k.off flat.140420.k

3. Normalize flatfield to ~ 1.0 using a reasonably large subregion:
   imstat flat.140420.k[500:1500,500:1500]; imar flat.140420.k / <stat> flat.140420.k

4. Set pixels below a certain value (say 0.05) to 1.0 to avoid excessively large numbers in the flattened image using imreplace (Fig. 2.5). Alternatively, ensure that a bad pixel map includes low-valued pixels.

5. Copy flats to a "fixed" extension: imcopy flat.140420.*.fits flat.140420.*.%fits%fx.fits% and run fixpix on these images (Fig. 2.9).

6. Generate a rough image of the pupil ghost by dividing the processed Ks flat by the J flat: imar flat.140420.fx.k / flat.140420.fx.j flatrat.kj

7. Knock down some of the bright spots in the ratio by running imreplace with a lower limit set just above the highest value in the pupil ghost: imcopy flatrat.kj flatrat.kj.edit; imreplace flatrat.kj.edit value=1 lower=1.2 upper=INDEF

8. Run fmedian (Fig. 2.10) to produce the smoothed image flatrat.kj.med

9. Subtract the median/mode from flatrat.kj.med to make the background ~ 0: imstat flatrat.kj.med ; imar flatrat.kj.med - <stat> flatrat.kj.med

10. Generate a null image: mkimage whircpupil.fits make 0 2 "2048 2048"

11. Copy the central 800 × 800 pixels of the pupil image into the null image: imcopy flatrat.kj.med[624:1424,624:1424] whircpupil[624:1424,624:1424]. This should give you an image like Fig. 2.11b.

12. Run rmpupil (Fig. 2.13) on the processed flat (e.g., flat.140420.fx.k) using the pupil image generated by fmedian as the pupil input. The result should look like Fig. 2.14b.

2.4.4 Fringes

For at least three of the WHIRC filters (Paβ, Paβ45, and H), Newton's Rings fringes have been seen in on-sky images. These almost certainly arise from OH emission lines in the atmosphere and are thus not seen in dome flats (although they may appear in H band sky flats). The optical configuration within WHIRC which produces these fringes is not known, but they are textbook Newton's Rings centered at the same location on the detector in all filters where they are seen. The Paβ and Paβ45 fringes appear monochromatic, whereas the H band fringes show amplitude modulation suggesting several wavelengths are contributing (most likely the strong OH Q-branch lines in the H band).

Unfortunately, unlike the pupil ghost, the fringe patterns differ for each filter in both spacing (wavelength-dependent) and phase. In addition, fringe patterns seen in a given filter appear to vary slightly in phase from one night to another. Therefore, one cannot use a generic template to attempt removal of the fringe pattern, but must use the data from the same night.

Because the strength of the OH lines varies during the night, fringes may not be seen in the sky frame (see section 3.1) for any given target, but they may very well appear in a "supersky" generated from all of the observations during a given night. This is often the case in the H band; in the Paβ and Paβ45 filters, the fringes will almost always be prominent since the continuum sky background is negligible and science exposures tend to be long.

The MOSAIC reduction task mscred.rmfringe appears to do a credible job of removing these fringes from the sky frame. As with the rmpupil task, one needs a template of the pattern, which will then be amplitude fitted to the fringes in the data and subtracted. Since a generic template cannot be used, one must generate the template from the science observations. *Refer to sections 3.1 and 3.2 following, since the fringe removal will be carried out during these steps of the data reduction*. It is logical to apply the fringe removal to the sky-subtracted data, since the sky subtraction may remove most, if not all, of the fringing, and it is generally preferable to reduce any artifacts to the extent possible before applying additional reduction steps.

Generating the fringe template is similar to making a sky flat (without the flat), so one will require, in addition to the science data, a dome flat in the same filter and *a series of darks taken at the same integration time as the science data*.

1. Create a sky image using imcombine on all of the images in a given filter on each target field with median filtering (Fig. 2.15). Since the sky level will probably be changing, one should scale the image using the median value of a large subregion such as [500:1500,500:1500]. Ideally, this will result in a sky image from which all of the stars have been eliminated. This may not work in extremely crowded fields. For very crowded fields or those containing an extended source, one should take dedicated sky observations (see section 3.1). If one observed several fields through the filter at the same integration time, one may generate a "master" sky image by imcombining all of these images; this may enhance the fringe visibility as well as increasing the S/N of the resulting frame. The result will look something like Fig. 2.16a.

2. Average the dark frames (as in making a sky flat, section 2.3) and subtract from the average sky frame (Fig. 2.16b).

joyce@fungo:~/iraf
IRAF
Image Reduction and Analysis Facility
PACKAGE = immatch
TASK = imcombine
input = @inlist List of images to combine
output = sky.ks List of output images
(headers=) List of header files (optional)
(bpmasks=) List of bad pixel masks (optional)
(rejmask=) List of rejection masks (optional)
(nrejmas=) List of number rejected masks (optional)
(expmask=) List of exposure masks (optional)
(sigmas =) List of sigma images (optional)
(imcmb =) Keyword for IMCMB keywords
(logfile= STDOUT) Log file
(combine= median) Type of combine operation
(reject = avsigclip) Type of rejection
(project= no) Project highest dimension of input images?
(outtype= real) Output image pixel datatype
(outlimi=) Output limits (x1 x2 y1 y2 ...)
(offsets= none) Input image offsets
(masktyp= none) Mask type
(maskval= 0) Mask value
(blank = 0.) Value if there are no pixels
(scale = median) Image scaling
(zero = none) Image zero point offset
(weight = none) Image weights
(statsec= [500:1500,500:1500]) Image section for computing statistics
(expname=) Image header exposure time keyword
(lthresh= INDEF) Lower threshold
(hthresh= INDEF) Upper threshold
(nlow = 1) minmax: Number of low pixels to reject
(nhigh = 1) minmax: Number of high pixels to reject
(nkeep = 1) Minimum to keep (pos) or maximum to reject (neg)
(mclip = yes) Use median in sigma clipping algorithms?
(lsigma = 3.) Lower sigma clipping factor
(hsigma = 3.) Upper sigma clipping factor
(rdnoise= 40) ccdclip: CCD readout noise (electrons)
(gain = 9) ccdclip: CCD gain (electrons/DN)
(snoise = 0.) ccdclip: Sensitivity noise (fraction)
(sigscal= 0.1) Tolerance for sigma clipping scaling corrections
(pclip = -0.5) pclip: Percentile clipping parameter
(grow = 0.) Radius (pixels) for neighbor rejection
(mode = ql)

Figure 2.15: IRAF parameter list for the task imcombine to obtain a sky image for the set of images in the file 'inlist'. Note the use of 'combine' = 'median' to remove the stars from the dithered images. This sky frame can be used for sky subtraction, generating a sky flat, or generating a fringe template. The process is similar for generating a dark image from a series of images, except the scale should be set to 'none' since the signal levels are very low.

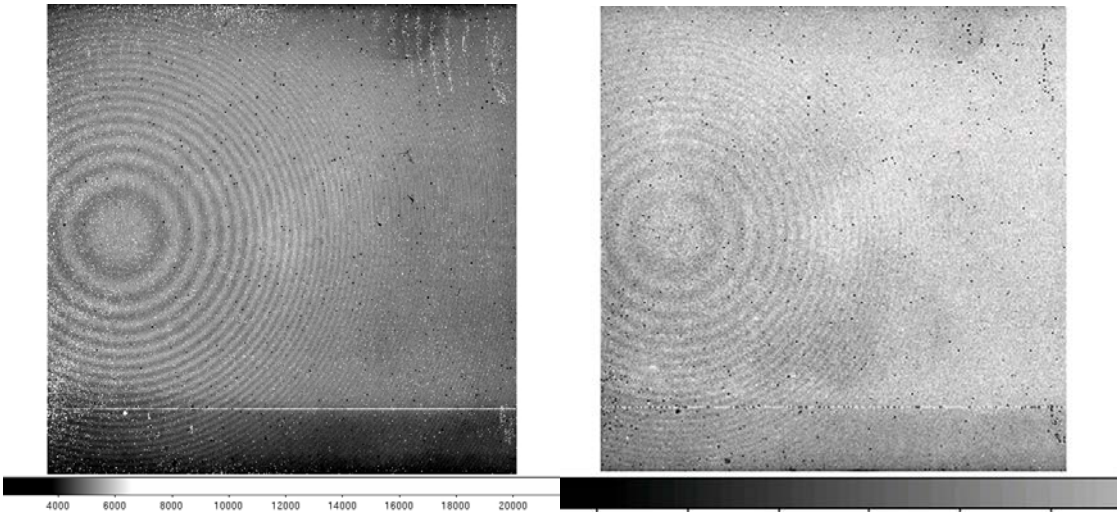

Figure 2.16: Average of sky images in the Paβ filter (left panel), showing evident fringes. Same image after subtraction of dark frame (right panel).

3. At this point most of the dark current will be removed, except for occasional mavericks, but the frame still has the low spatial frequency structure and the central pupil ghost. Dividing by the dome flat and running fixpix with the bad pixel mask (Fig. 2.9) will remove most of the structure and clean up the image (Fig. 2.18a).

4. The fringe template for rmfringe has (ideally) a minimum value near zero, so one can at this point subtract off a constant value to achieve this. Running fmedian (Fig. 2.17) with a fairly small box (3 × 3) appears to enhance the S/N without affecting the spatial resolution of the template (Fig. 2.18b).

```
joyce@fungo:~/iraf
                                    IRAF
                Image Reduction and Analysis Facility
PACKAGE = imfilter
   TASK = fmedian

input   =          fringe.pab  Input images to be filtered
output  =       fringe.pab.sm  Output images
xwindow = |                3  X window size of median filter
ywindow =                  3  Y window size of median filter
(hmin   =             -32768) Minimum histogram bin
(hmax   =              32767) Maximum histogram bin
(zmin   =              INDEF) Pixel value corresponding to hmin
(zmax   =              INDEF) Pixel value corresponding to hmax
(zloreje=              INDEF) Lowside pixel value cutoff
(zhireje=              INDEF) High side pixel value cutoff
(unmap  =                yes) Unmap the digitized values ?
(boundar=           nearest) Boundary (constant,nearest,reflect,wrap)
(constan=                0.) Constant for boundary extension
(verbose=                yes) Print messages about actions taken by the task
(mode   =                 ql)
```

Figure 2.17: IRAF parameter list for the task fmedian, with a smaller (3 × 3) smoothing box than used for the pupil ghost template.

For comparison, we show in Fig. 2.19 fringe templates for the Paβ45 and H filters, to show the need for filter and time-specific fringe information. While the Paβ45 pattern seems similar to the Paβ pattern, one can see a phase difference in the central fringe. The fringe visibility in the H band varies, suggesting that several emission lines contribute to the fringes.

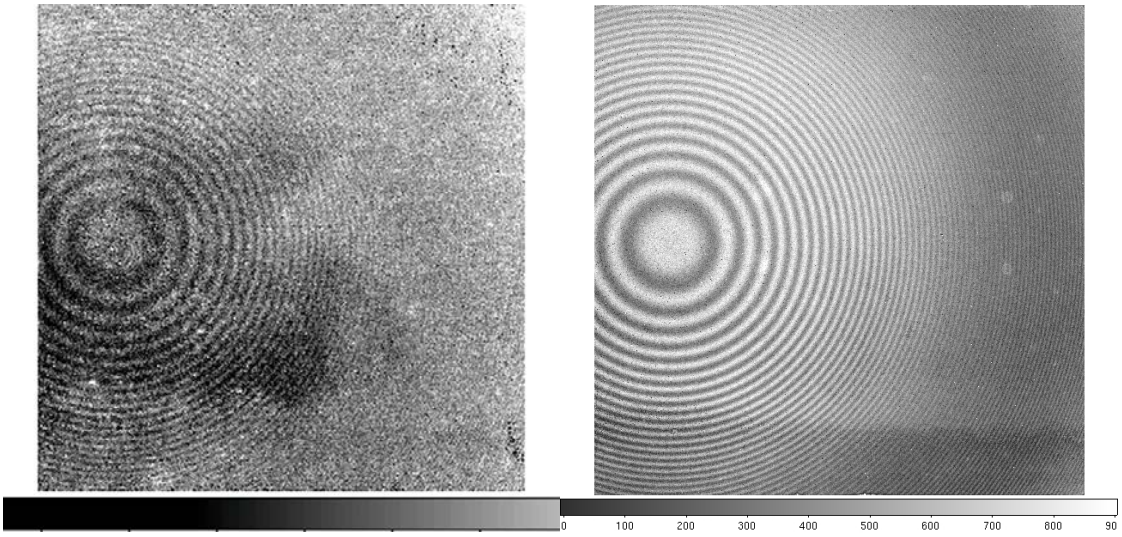

Figure 2.18: (left panel) Paβ sky image from Fig. 2.16b after dividing by a dome flat and cleaning up bad pixels. Some of the "palmprint" can still be seen. (right panel): After subtracting the background and smoothing, one has an adequate fringe template. The white blobs are probably artifacts from bright stars in the original images which were not completely removed by the median filtering.

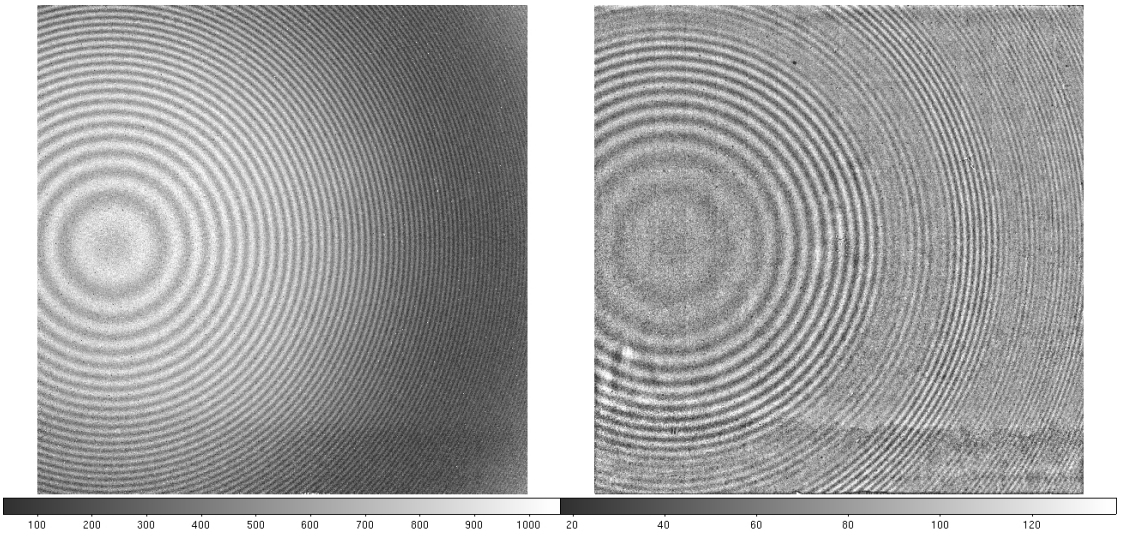

Figure 2.19: Fringe templates derived from sky data in the Paβ45 (left panel) and H (right panel) filters. Note the phase shift in the central fringe between the Paβ (Fig. 2.18b) and Paβ45 templates and the amplitude modulation in the H template.

5. Using the appropriate fringe template, the task rmfringe can then be used to remove the fringes (Fig. 2.20). Figure 2.21 illustrates this for a H band sky image on a night where fringing was evident. In practice, one would operate on a sky-subtracted image, where the fringe pattern should be much less evident.

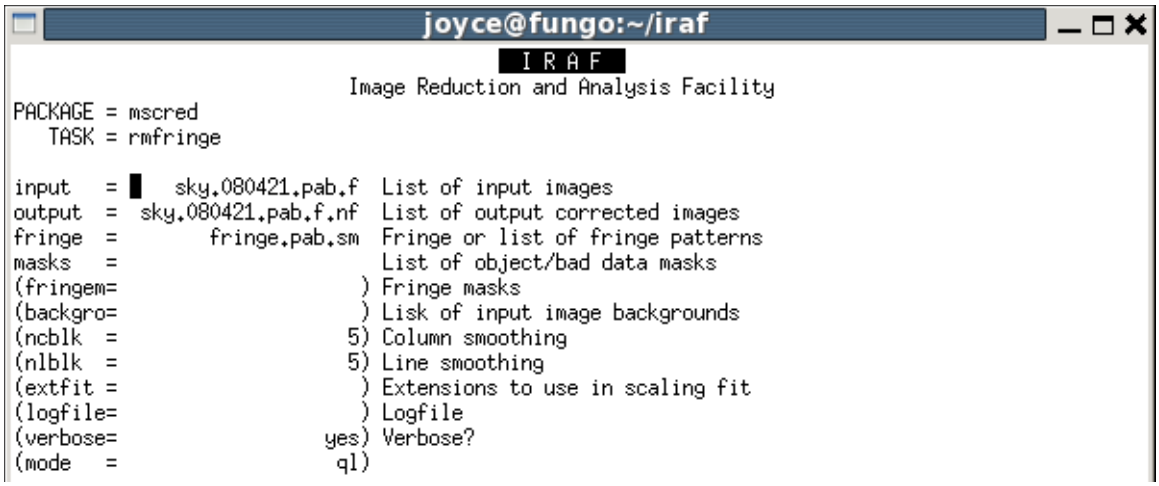
joyce@fungo:~/iraf
IRAF
Image Reduction and Analysis Facility
PACKAGE = mscred
TASK = rmfringe
input = sky.080421.pab.f List of input images
output = sky.080421.pab.f.nf List of output corrected images
fringe = fringe.pab.sm Fringe or list of fringe patterns
masks = List of object/bad data masks
(fringem=) Fringe masks
(backgro=) Lisk of input image backgrounds
(ncblk = 5) Column smoothing
(nlblk = 5) Line smoothing
(extfit =) Extensions to use in scaling fit
(logfile=) Logfile
(verbose= yes) Verbose?
(mode = ql)

Figure 2.20: IRAF parameter listing for the task rmfringe using the smoothed Paβ fringe pattern from Fig. 2.17b.

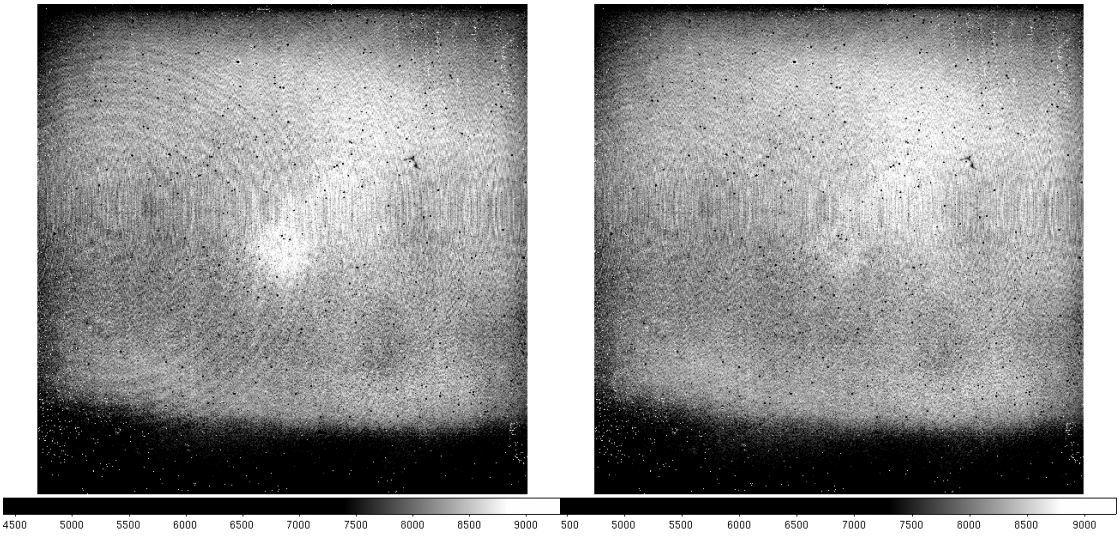

Fig. 2.21: H band sky image (left panel) on a night when fringing was evident. After processing with the rmfringe task, the fringes are almost completely eliminated (right panel).

3.0 Data Reduction

3.1 Standards or Pointlike Objects

1. One will generally have several images of the field with different telescope pointings. Depending on the sky stability, the sky level may be somewhat different in each of the images.

2. Create a sky image using imcombine on all of the images in a given filter on each target field with median filtering, as described in section 2.4.4. Since the sky level will probably be changing, one should scale the image using the median value of a large subregion such as [500:1500,500:1500]. Ideally, this will result in a sky image from which all of the stars have been eliminated. This may not work in extremely crowded fields. For very crowded fields or those containing an extended source, one should take dedicated sky observations (see below). See parameter file in Fig. 2.15.

3. Inspect the sky image and if it looks OK, generate sky-subtracted images by subtracting the sky image from each of the original images. If the sky level had changed during the observation series, some of the sky levels may be positive, some negative. Keep going. Fringe removal, if necessary, can be done at this stage.

4. Divide each sky-subtracted image by the flat for that filter. This should result in images in which the (non-zero) sky level is uniform across the image. One can subtract this offset out by measuring the median level of the image and subtracting that number. If one is doing aperture photometry, the sky level should be automatically subtracted during that process.

5. One may analyze the results from each image separately or combine them, using upsqiid or other custom routines. If making a large mosaic of images, distortion correction will be needed to ensure good overlap of the stars in different images (section 3.3.1).

3.2 Extended Objects or Crowded Fields

1. As with the pointlike fields, one should have several dithered images of the science field, but also a set of dithered images taken at a not too distant “sky” position relatively devoid of stars.

2. Combine the off-target sky images as above using median filtering.

3. Subtract the sky image from each of the target images to yield the sky-subtracted image.

4. Flatfield the sky-subtracted images as above.

5. One may analyze the results from each image separately or combine them, using upsqiid or other custom routines.

3.3 Combining Images into a Mosaic

For IRAF users, the upsqiid package written by Mike Merrill can be used to align and combine the images into a single image. This is primarily useful for deep imaging, where one is observing the same field with relatively small dither amplitudes, since it relies on locating star(s) common to all images for alignment. Although this package was written primarily for data from the 4-color imager SQIID, the routines for aligning and combining individual images into a mosaic are applicable for WHIRC images. A description of the upsqiid package, as well as instructions for downloading the package into IRAF, can be found at http://www.noao.edu/kpno/sqiid/upsqiidpkg.html. There are also more generic IRAF tasks in the noao.imred.irred package which can be used for larger mosaics/maps.

One may consider three different scenarios for combining individual images:

1. Imaging of point or small sources, where one is concerned only with a limited portion of the WHIRC field.

2. Imaging of extended sources or fields which occupy a significant (> 50%) portion of the WHIRC field.

3. Imaging of extended sources or fields which are significantly larger than the WHIRC field (sometimes referred to as mapping).

The first two of these may be handled using the upsqiid tasks, with the important provision that the dithering amplitude used during data acquisition be sufficiently small that there exists an alignment star common to *all* of the image fields. The difference between the two scenarios is that the second will require correction of the optical distortion of the WHIRC image (*vide infra*) to yield good superposition of the shifted images. The third scenario will require use of the IRAF noao.imred.irred tasks, but will still require common stars in all *adjacent* frames.

3.3.1 Distortion Correction

Because WTTM utilizes off-axis reflective optics, the plate scale is different in the X and Y coordinates on the detector (0.0969 and 0.1002 arcsec/pixel, respectively). In addition, there is a small amount of field distortion at the input to WHIRC. The distortion is asymmetric (keystone) and is less than 1%. The additional distortion from WHIRC is very small and is negligibly dependent on wavelength.

During the design of WHIRC, the entire optical system (WIYN telescope, WTTM, and WHIRC) was modeled using Zemax, and optical distortion maps were generated in the J, H, and Ks filters. These files were used as input to the IRAF routine geomap to generate third-order polynomial fits with an rms error ~ 0.1 pixel. The files whirc.distort.<filter>.txt and whirc.distort.<filter>.db produced by geomap can be downloaded from the WHIRC webpage and used in the routine geotran to produce distortion-corrected images which can then be combined into a large mosaic with good registration to the edges of the field (Fig. 3.8). The chromatic effects on distortion are extremely small, so the distortion files are almost identical.

NOTE: Because geotran remaps each pixel slightly, one should carry out the distortion correction on images which have already been processed by sky subtraction and flatfielding. We have not investigated the effects of distortion correction on surface brightness measurements of extended sources. The geotran routine *should* preserve flux when the 'fluxconserve' flag is set to 'yes'. We suggest that observers be conservative, or at least cognizant, of the potential effects of distortion correction on photometry. Programs utilizing deep imaging of a small field using small dither amplitudes (scenario 1 above) may not require correction. Imaging over a more extended field (scenario 2 above), even with small dither amplitudes, may benefit from distortion correction when using PSF-fitting photometry.

For large scale mosaics/maps (scenario 3 above) it is necessary to carry out the distortion correction prior to proceeding to the mosaic to ensure that stars in adjacent images will be properly superposed. For the smaller scale mosaics (scenarios 1 and 2 above), one may carry out the distortion correction separately or within the alignment task.

The geotran task parameter set is illustrated in Figure 3.1. In this example, the input images have been put into a list (clist) and the output images into a separate list (dlist). For example, clist may consist of images such as 'M13.h.f.1,2,3,…' and dlist of images such as 'M13.h.geo.1,2,3…' The task will create the new images. The database and transform files will be the 'whirc.distort.<filter>.db' and "whirc.distort.<filter>.txt' files referred to above.

If one displays uncorrected and corrected images, particularly in "blink" mode, it is possible to see the difference, which amounts to almost 10 pixels (1 arcsec) at the periphery of the image. It is easy to appreciate the poor results which one would obtain if attempting to combine images without the geometric correction.

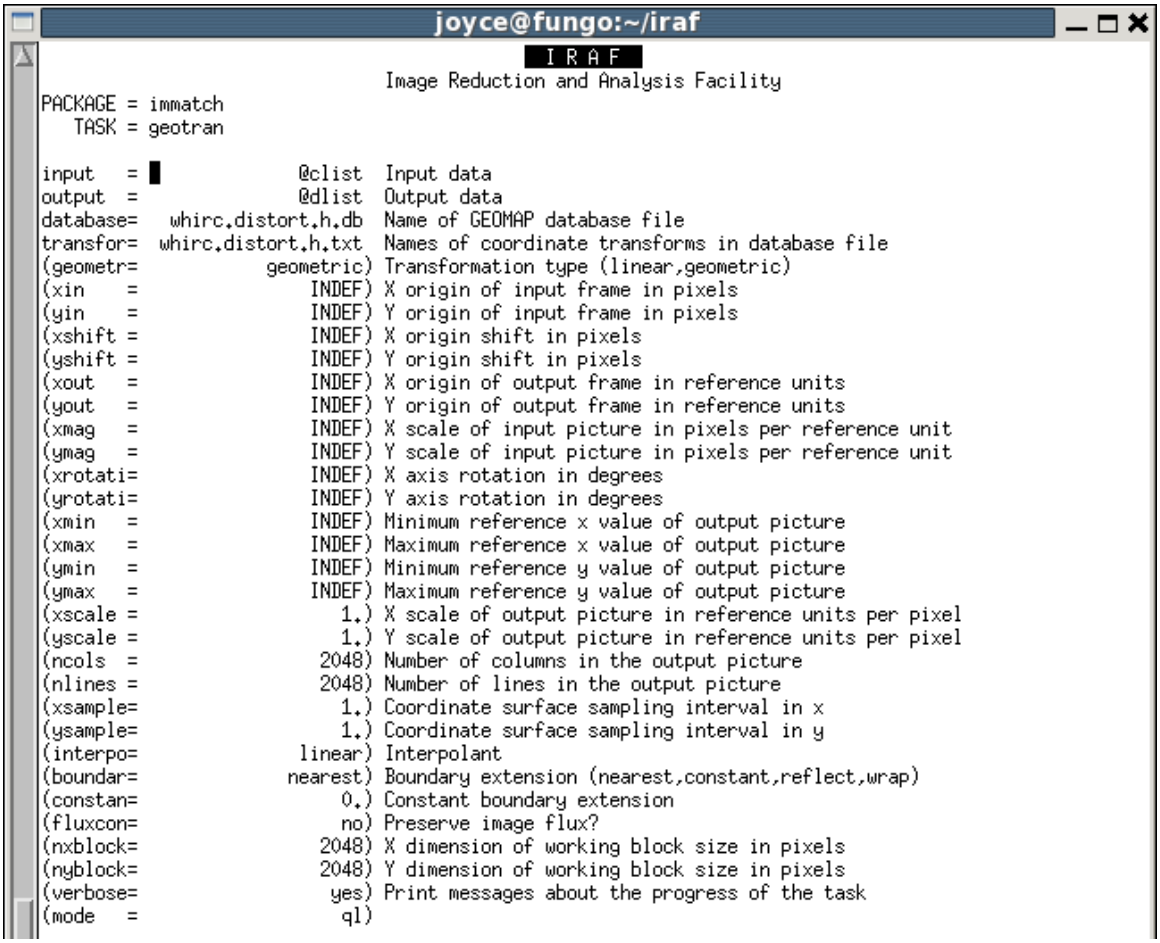
joyce@fungo:~/iraf
IRAF
Image Reduction and Analysis Facility
PACKAGE = immatch
TASK = geotran
input = @clist Input data
output = @dlist Output data
database= whirc.distort.h.db Name of GEOMAP database file
transfor= whirc.distort.h.txt Names of coordinate transforms in database file
(geometr= geometric) Transformation type (linear,geometric)
(xin = INDEF) X origin of input frame in pixels
(yin = INDEF) Y origin of input frame in pixels
(xshift = INDEF) X origin shift in pixels
(yshift = INDEF) Y origin shift in pixels
(xout = INDEF) X origin of output frame in reference units
(yout = INDEF) Y origin of output frame in reference units
(xmag = INDEF) X scale of input picture in pixels per reference unit
(ymag = INDEF) Y scale of input picture in pixels per reference unit
(xrotati= INDEF) X axis rotation in degrees
(yrotati= INDEF) Y axis rotation in degrees
(xmin = INDEF) Minimum reference x value of output picture
(xmax = INDEF) Maximum reference x value of output picture
(ymin = INDEF) Minimum reference y value of output picture
(ymax = INDEF) Maximum reference y value of output picture
(xscale = 1.) X scale of output picture in reference units per pixel
(yscale = 1.) Y scale of output picture in reference units per pixel
(ncols = 2048) Number of columns in the output picture
(nlines = 2048) Number of lines in the output picture
(xsample= 1.) Coordinate surface sampling interval in x
(ysample= 1.) Coordinate surface sampling interval in y
(interpo= linear) Interpolant
(boundar= nearest) Boundary extension (nearest,constant,reflect,wrap)
(constan= 0.) Constant boundary extension
(fluxcon= no) Preserve image flux?
(nxblock= 2048) X dimension of working block size in pixels
(nyblock= 2048) Y dimension of working block size in pixels
(verbose= yes) Print messages about the progress of the task
(mode = ql)

Figure 3.1: Parameter file for the geotran task.

3.3.2 Small Field Mosaics

The two basic tasks within the upsqiid package which are useful for combining images:

xyget – Find common stars in the images and create a registration database
nircombine – Combine the registration database into a composite image

3.3.2.1 The xyget Task

The task xyget is an interactive task in which each of the images is displayed in turn and a star common to all of the images is identified on the display. The measured centroid of each of the star images is put into a database file of the sort '<inlist>.xycom', where 'inlist' is the list of input images. The '<inlist>.xycom' file is then used as the input to the nircombine task which will combine the input images into a single output image.

1. Reduce the images to be mosaicked (sky subtraction, flatfielding, etc.) as above and put them into a list.

2. I usually like to set the sky levels for all of the images to zero before running xyget, but this is optional, since one can also carry out this step automatically within the nircombine task. (Although the two approaches seem to give slightly different sky levels in the combined image). One needs to set the imstat task so that the mode (or midpoint) is one of the output columns:
    a. imstat @inlist > statlist (runs imstats and puts results in file statlist)
    b. fields statlist 4 > sslist (writes column 4 or whatever mode is, into list sslist)
    c. imar @inlist - @sslist @inlist (subtracts the mode from each image)

3. Run the xyget task. Figure 3.2 illustrates the parameter set for xyget *without* distortion correction, and Figure 3.3 illustrates the parameter set *with* distortion correction. The only difference is that the parameter 'tran' is set to 'no' in the first case and to 'yes' in the second.
    a. The 'images' parameter is the list containing the images to be analyzed
    b. The 'frame_nu' parameter is the number of images to be analyzed and the initial reference image. In this case, there are five images in the list, all of which will be measured, and the first will be used as the reference.
    c. The 'bigbox' parameter is the initial search box size, and 'boxsize' is the final centering box. The numbers given seem appropriate for reasonable seeing with WHIRC. However, if the output file '<list>.xycom' is missing one or more of the images (i.e., couldn't centroid on the star), one can make the 'boxsize' slightly larger; this can happen if the seeing is not as good.
    d. When you run xyget, the first image in the list will be displayed on the image display. Identify a star in the field which is well isolated, unsaturated, and is known to be in all of the images, put the cursor on it, and hit <spacebar>. The star coordinates will be displayed in the IRAF xgterm window.
    e. One star is usually sufficient for alignment, so one can enter 'q' to stop this. The program will return "Do you want to get frame offsets? (yes)". Answer affirmatively with <CR>.
    f. The program will return "Select star for reference frame 1". You can just hit the <spacebar> on the same star. The program will then display frame 1 in the list, which will be the same image again, if you selected frame 1 as the reference in the 'frame_nu' parameter.
    g. Center the cursor on the same star and hit <spacebar>. The succeeding images will be displayed in turn; move the cursor to the same star in each frame and hit <spacebar>. After the last image, the program will display a summary of the centroid information (Figure 3.4). Check to make sure that all images have output data. If one or more is missing from the list, you may need to adjust the 'boxsize' parameter and re-run xyget. In this case you will have to delete the '<list>.xycom' file.
    h. I usually like to rename the '<list>.xycom' file to something unique (e.g., 'M13.h.xycom') so that it is not overwritten by succeeding runs of xyget.

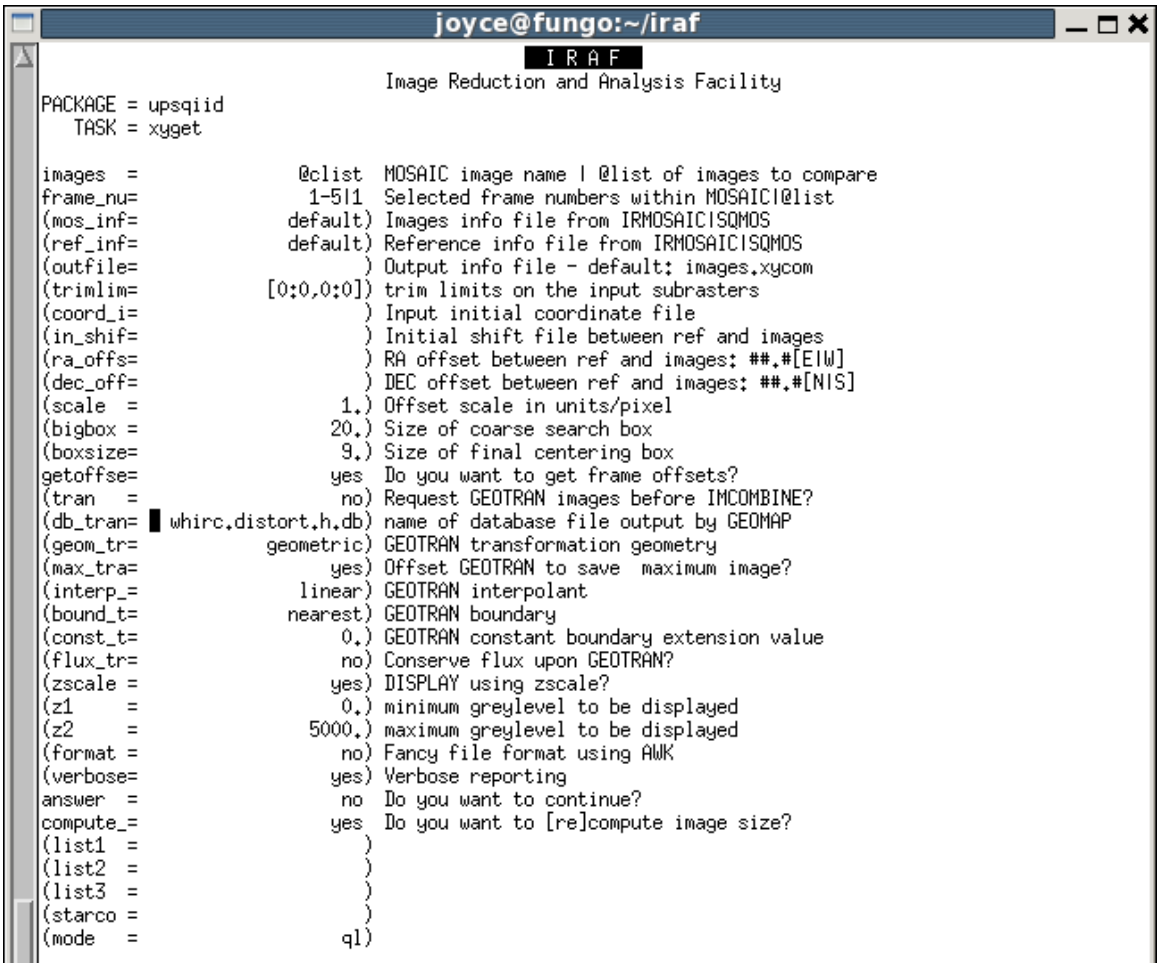
joyce@fungo:~/iraf
IRAF
Image Reduction and Analysis Facility
PACKAGE = upsqiid
TASK = xyget
images = @clist MOSAIC image name | @list of images to compare
frame_nu= 1-5|1 Selected frame numbers within MOSAIC|@list
(mos_inf= default) Images info file from IRMOSAIC|SQMOS
(ref_inf= default) Reference info file from IRMOSAIC|SQMOS
(outfile=) Output info file - default: images.xycom
(trimlim= [0:0,0:0]) trim limits on the input subrasters
(coord_i=) Input initial coordinate file
(in_shif=) Initial shift file between ref and images
(ra_offs=) RA offset between ref and images: ##.#[EW]
(dec_off=) DEC offset between ref and images: ##.#[NS]
(scale = 1.) Offset scale in units/pixel
(bigbox = 20.) Size of coarse search box
(boxsize= 9.) Size of final centering box
getoffse= yes Do you want to get frame offsets?
(tran = no) Request GEOTRAN images before IMCOMBINE?
(db_tran= whirc.distort.h.db) name of database file output by GEOMAP
(geom_tr= geometric) GEOTRAN transformation geometry
(max_tra= yes) Offset GEOTRAN to save maximum image?
(interp_= linear) GEOTRAN interpolant
(bound_t= nearest) GEOTRAN boundary
(const_t= 0.) GEOTRAN constant boundary extension value
(flux_tr= no) Conserve flux upon GEOTRAN?
(zscale = yes) DISPLAY using zscale?
(z1 = 0.) minimum greylevel to be displayed
(z2 = 5000.) maximum greylevel to be displayed
(format = no) Fancy file format using AWK
(verbose= yes) Verbose reporting
answer = no Do you want to continue?
compute_= yes Do you want to [re]compute image size?
(list1 =)
(list2 =)
(list3 =)
(starco =)
(mode = ql)

Figure 3.2: Parameter set for the task xyget without distortion correction. The parameter 'tran' is set to 'no'.

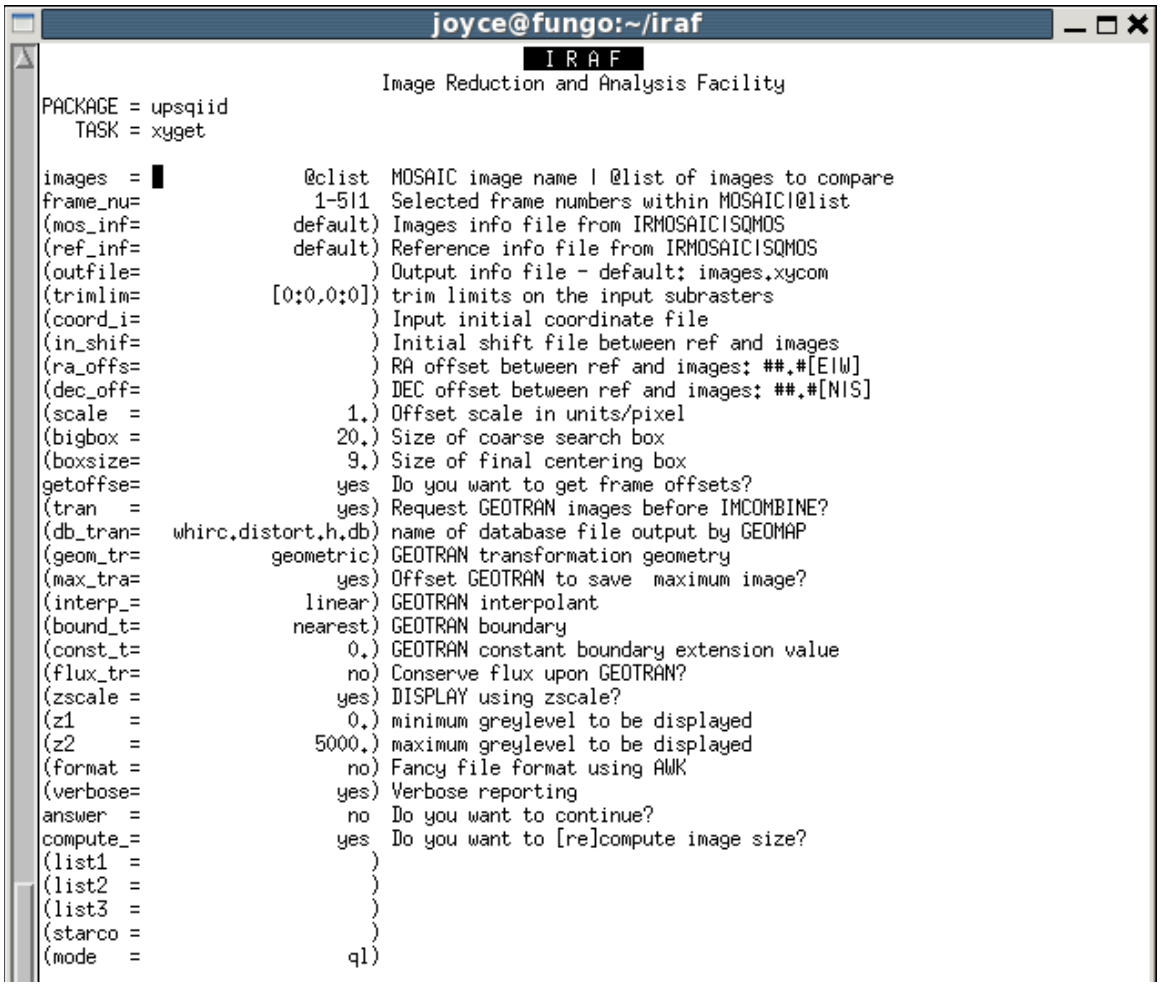
joyce@fungo:~/iraf
IRAF
Image Reduction and Analysis Facility
PACKAGE = upsqiid
TASK = xyget
images = @clist MOSAIC image name | @list of images to compare
frame_nu= 1-5|1 Selected frame numbers within MOSAIC|@list
(mos_inf= default) Images info file from IRMOSAIC|SQMOS
(ref_inf= default) Reference info file from IRMOSAIC|SQMOS
(outfile=) Output info file - default: images.xycom
(trimlim= [0:0,0:0]) trim limits on the input subrasters
(coord_i=) Input initial coordinate file
(in_shif=) Initial shift file between ref and images
(ra_offs=) RA offset between ref and images: ##.#[E|W]
(dec_off=) DEC offset between ref and images: ##.#[N|S]
(scale = 1.) Offset scale in units/pixel
(bigbox = 20.) Size of coarse search box
(boxsize= 9.) Size of final centering box
getoffse= yes Do you want to get frame offsets?
(tran = yes) Request GEOTRAN images before IMCOMBINE?
(db_tran= whirc.distort.h.db) name of database file output by GEOMAP
(geom_tr= geometric) GEOTRAN transformation geometry
(max_tra= yes) Offset GEOTRAN to save maximum image?
(interp_= linear) GEOTRAN interpolant
(bound_t= nearest) GEOTRAN boundary
(const_t= 0.) GEOTRAN constant boundary extension value
(flux_tr= no) Conserve flux upon GEOTRAN?
(zscale = yes) DISPLAY using zscale?
(z1 = 0.) minimum greylevel to be displayed
(z2 = 5000.) maximum greylevel to be displayed
(format = no) Fancy file format using AWK
(verbose= yes) Verbose reporting
answer = no Do you want to continue?
compute_= yes Do you want to [re]compute image size?
(list1 =)
(list2 =)
(list3 =)
(starco =)
(mode = ql)

Figure 3.3: Parameter set for the task xyget with distortion correction. The parameter 'tran' is set to 'yes'.

```
IMCENTROID: @tmp$gcm3762we
Will [re]optimize origin.
-514 2564 -503 2538
APPLIED_OFFSETS: 514. 503.
ENCLOSED_REGION: [0:3078,0:3041]
ENCLOSED_SIZE: [1:3079,1:3042]
UNAPPLIED_OFFSETS: 0. 0.
OVERLAP: [1031:2047,994:2047] [517:1533,491:1544]
COM_000 m13.h.p.1[1:2048,1:2048]   1 2048   1 2048  514  503  0.00  0.00     0.000
COM_001 m13.h.p.1[1:2048,1:2048]   1 2048   1 2048  514  503  0.00  0.00     0.000
COM_002 m13.h.p.2[1:2048,1:2048]   1 2048   1 2048  523    0 -0.12 -0.36     0.000
COM_003 m13.h.p.3[1:2048,1:2048]   1 2048   1 2048  505  992 -0.22  0.12     0.000
COM_004 m13.h.p.4[1:2048,1:2048]   1 2048   1 2048    0  488 -0.41 -0.22     0.000
COM_005 m13.h.p.5[1:2048,1:2048]   1 2048   1 2048 1030  501 -0.45 -0.47     0.000
#NOTE: overlap section: [1031:2047,994:2047]
vocl>
```

Figure 3.4: Output at completion of the xyget task. Note that all of the five images have offset data.

3.3.2.2 Combining Images with nircombine

The task nircombine (Figure 3.5) will combine the input images using the '<list>.xycom' output of xyget.

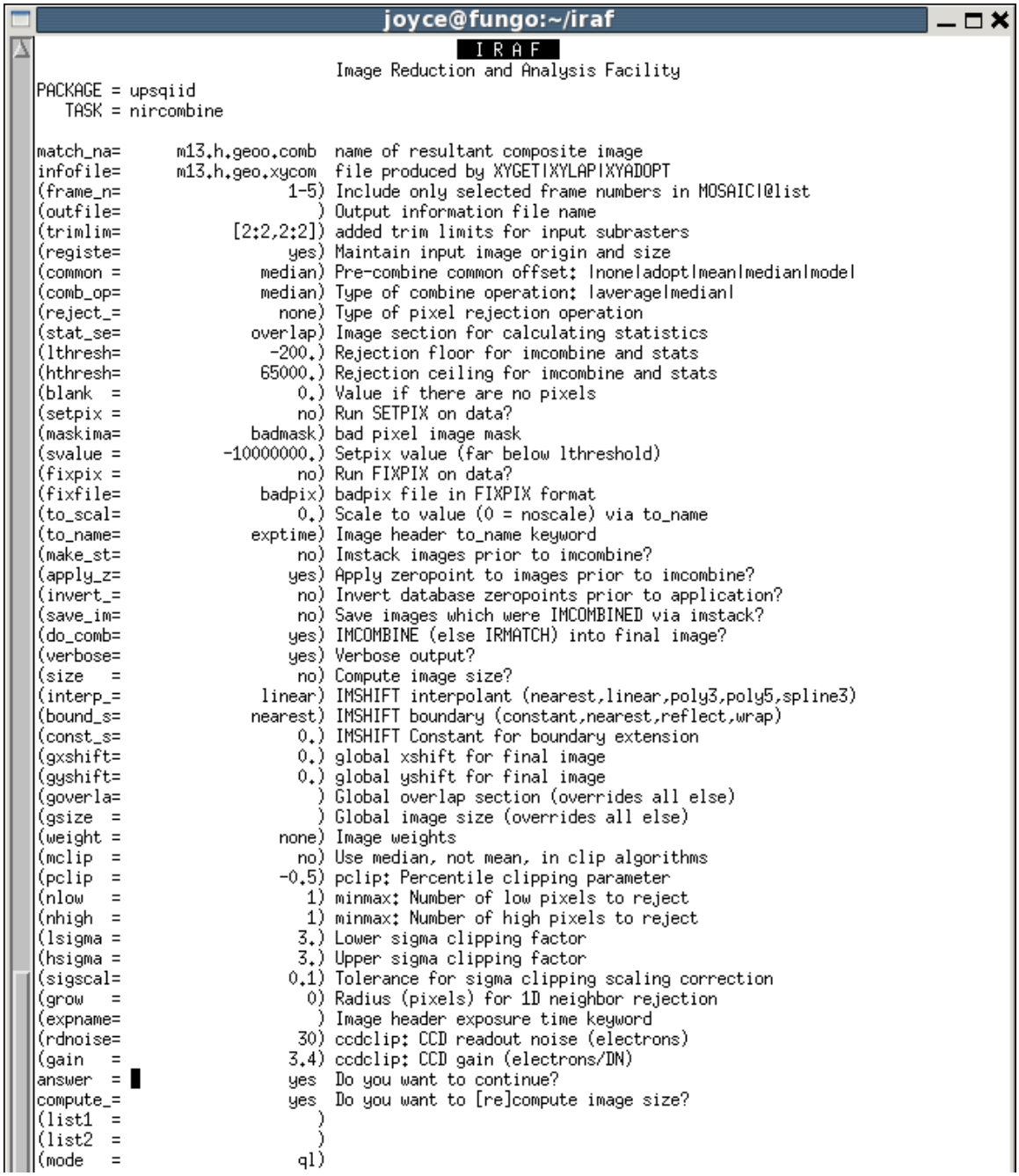
joyce@fungo:~/iraf
IRAF
Image Reduction and Analysis Facility
PACKAGE = upsqiid
TASK = nircombine
match_na= m13.h.geoo.comb name of resultant composite image
infofile= m13.h.geo.xycom file produced by XYGET|XYLAP|XYADOPT
(frame_n= 1-5) Include only selected frame numbers in MOSAIC|@list
(outfile=) Output information file name
(trimlim= [2:2,2:2]) added trim limits for input subrasters
(registe= yes) Maintain input image origin and size
(common = median) Pre-combine common offset: |none|adopt|mean|median|mode|
(comb_op= median) Type of combine operation: |average|median|
(reject_= none) Type of pixel rejection operation
(stat_se= overlap) Image section for calculating statistics
(lthresh= -200.) Rejection floor for imcombine and stats
(hthresh= 65000.) Rejection ceiling for imcombine and stats
(blank = 0.) Value if there are no pixels
(setpix = no) Run SETPIX on data?
(maskima= badmask) bad pixel image mask
(svalue = -10000000.) Setpix value (far below lthreshold)
(fixpix = no) Run FIXPIX on data?
(fixfile= badpix) badpix file in FIXPIX format
(to_scal= 0.) Scale to value (0 = noscale) via to_name
(to_name= exptime) Image header to_name keyword
(make_st= no) Imstack images prior to imcombine?
(apply_z= yes) Apply zeropoint to images prior to imcombine?
(invert_= no) Invert database zeropoints prior to application?
(save_im= no) Save images which were IMCOMBINED via imstack?
(do_comb= yes) IMCOMBINE (else IRMATCH) into final image?
(verbose= yes) Verbose output?
(size = no) Compute image size?
(interp_= linear) IMSHIFT interpolant (nearest,linear,poly3,poly5,spline3)
(bound_s= nearest) IMSHIFT boundary (constant,nearest,reflect,wrap)
(const_s= 0.) IMSHIFT Constant for boundary extension
(gxshift= 0.) global xshift for final image
(gyshift= 0.) global yshift for final image
(goverla=) Global overlap section (overrides all else)
(gsize =) Global image size (overrides all else)
(weight = none) Image weights
(mclip = no) Use median, not mean, in clip algorithms
(pclip = -0.5) pclip: Percentile clipping parameter
(nlow = 1) minmax: Number of low pixels to reject
(nhigh = 1) minmax: Number of high pixels to reject
(lsigma = 3.) Lower sigma clipping factor
(hsigma = 3.) Upper sigma clipping factor
(sigscal= 0.1) Tolerance for sigma clipping scaling correction
(grow = 0) Radius (pixels) for 1D neighbor rejection
(expname=) Image header exposure time keyword
(rdnoise= 30) ccdclip: CCD readout noise (electrons)
(gain = 3.4) ccdclip: CCD gain (electrons/DN)
answer = yes Do you want to continue?
compute_= yes Do you want to [re]compute image size?
(list1 =)
(list2 =)
(mode = ql)

Figure 3.5: Parameter set for the IRAF task 'nircombine'. The 'apply_z" parameter is set to 'yes' to normalize any offsets in the sky level of the input images.

a. The first parameter 'match_na' will be the name of the resultant image

b. The parameter ‘infofile’ is the ‘<list>.xycom’ output file from xyget.
c. The parameter ‘frame_n’ is the list of images one will combine. Normally, this would be the entire set of input images, but there may be reasons (bad seeing, telescope jump, etc.) why one may wish to exclude some images.
d. The parameter ‘comb_op’ specifies the method used for combining the images. Using a median combination is aesthetically pleasing because it usually results in the removal of fixed image artifacts which survive the bad pixel repairs earlier. However, if one is concerned about photometric accuracy, one may wish to average the images to avoid signal clipping which the median can introduce. One can run the task with both modes and compare the results.
e. The parameter ‘apply_z’ will normalize the sky levels in the images before combining them.

Figure 3.6 shows the results of the data reduction at various stages.

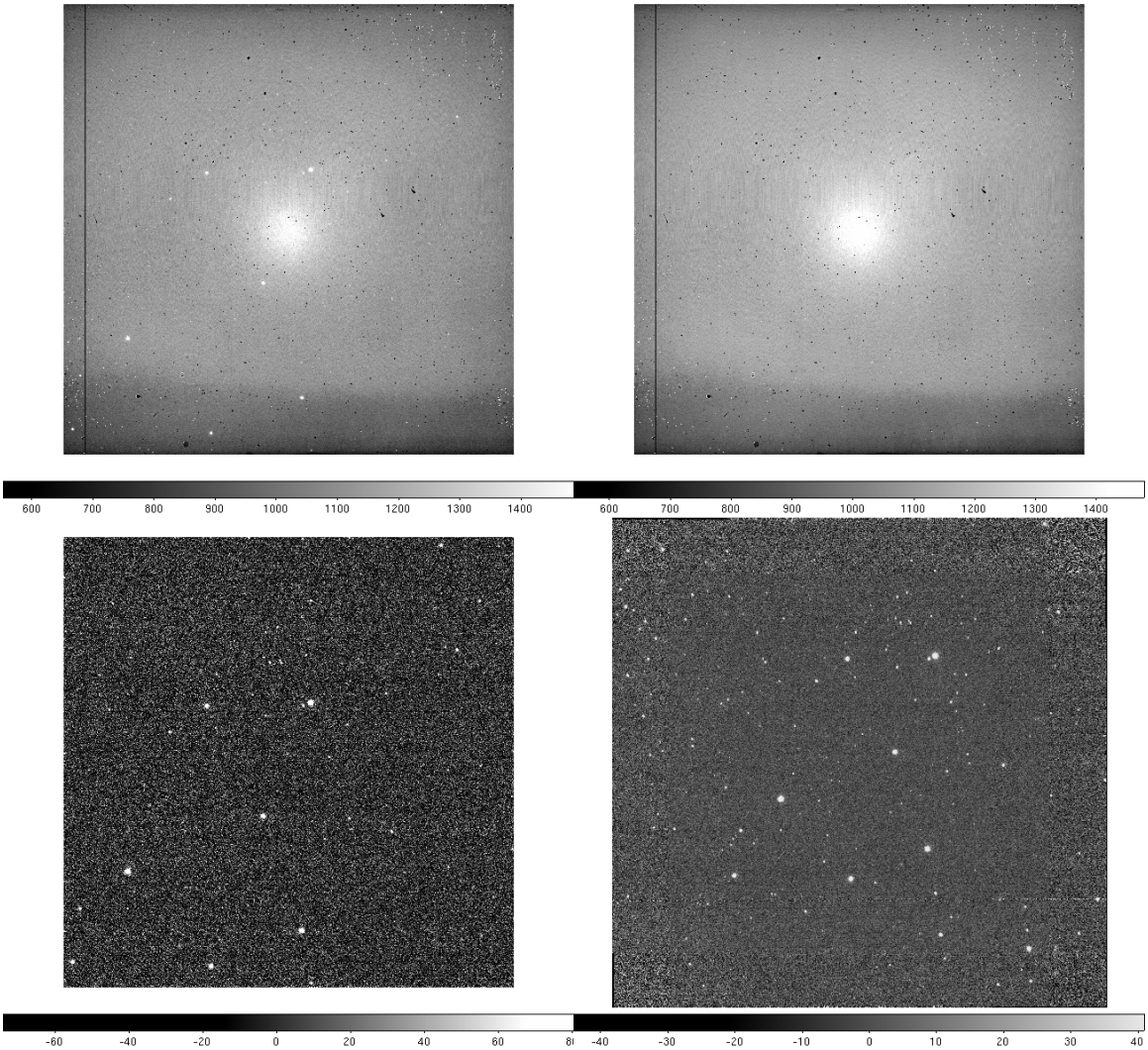

Figure 3.6: (upper left) Image of the NGC 7790 field in Ks, after trimming. This is one of nine images of the field, taken in a 3 × 3 grid with 50 arcsec spacing. (upper right) Sky image obtained by combining the nine input images with a median

filtering algorithm. (lower left) Difference of the top two images. Note that the sky subtraction reduces the residual background to near zero. (lower right) After flatfielding, the nine images are combined into a single composite using the upsqiid tasks xyget and nircombine. Note that the noise is higher in the periphery of the image where only a single image contributes to the mosaic.

Combining images with extended nebulosity may require some individualized fine tuning of the intensity levels, since the algorithm used in the nircombine task uses the statistics of the common overlap area. While this should in theory produce good matching at the periphery of overlapped regions, sometimes it is necessary to tweak the levels in the individual images to get a better match. Even with perfect matching of the intensity levels, one can detect the overlap transitions because the noise will be lower in the central parts of the mosaic, where more images were combined.

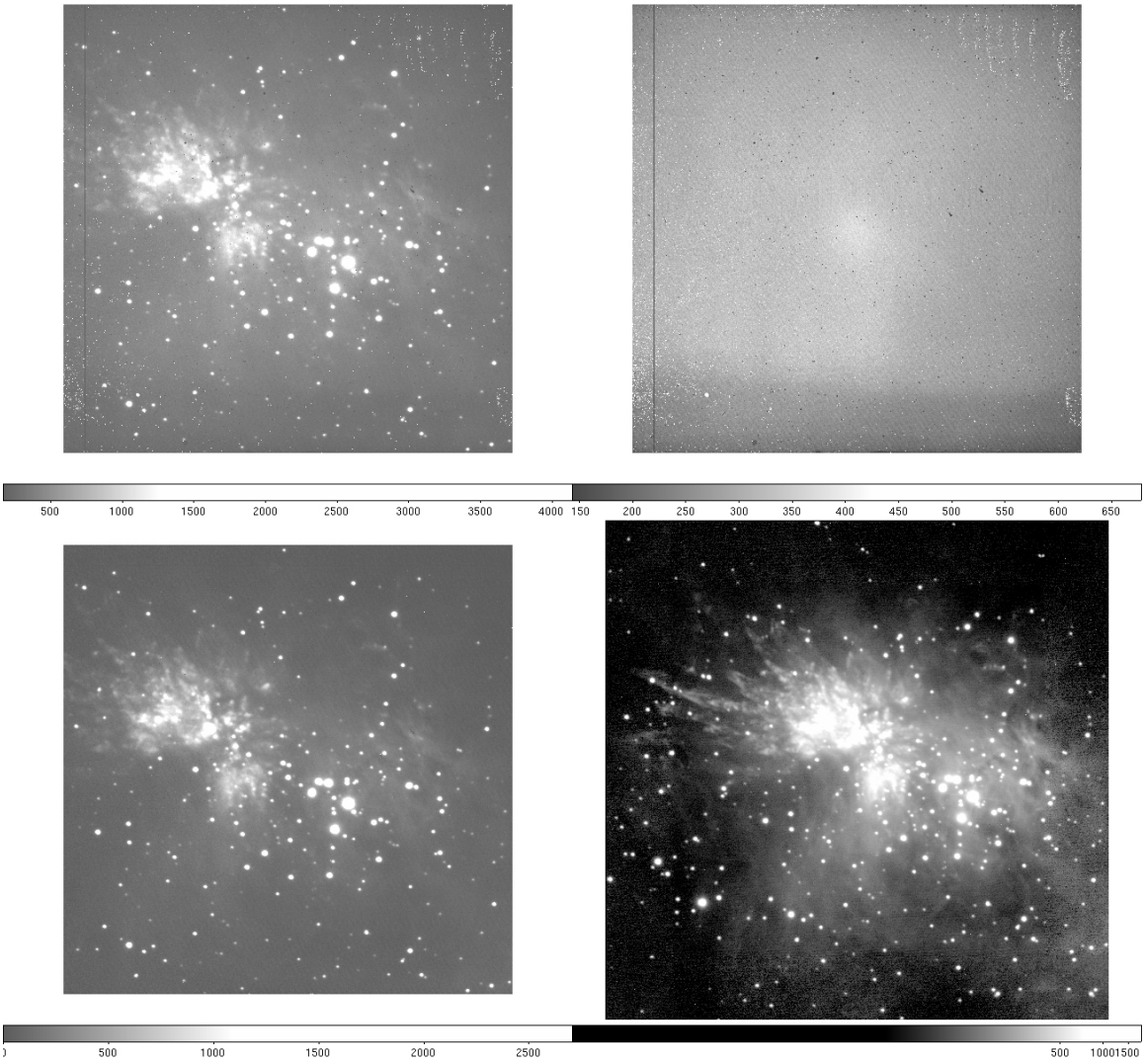

Figure 3.7: (upper left) Image of the M42 field in the narrowband H2 filter, after trimming. This is one of five images of the field, taken in a plus pattern with 50 arcsec spacing. (upper right) Sky image obtained by combining the five images of a

nearby sparse field with a median filtering algorithm. (lower left) Difference of the top two images. Note that the sky subtraction reduces the residual background to near zero. (lower right) After flatfielding, the five images are combined into a single composite using the upsqiid tasks xyget and nircombine. Note that the noise is higher in the periphery of the image where only a single image contributes to the mosaic. This image is displayed using a logarithmic scaling to bring out faint details. Observations courtesy M. Meixner.

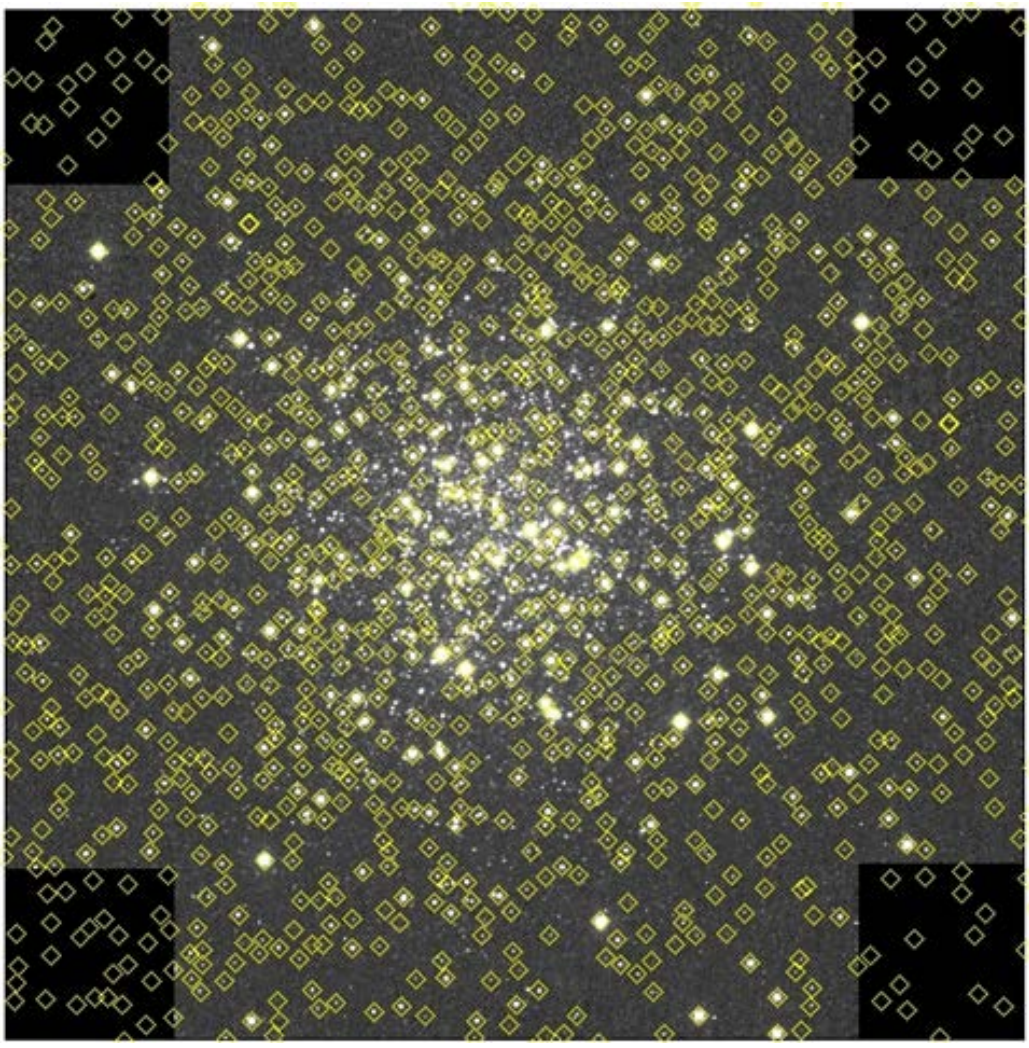

Figure 3.8: Mosaic of five images of the M13 field in the H band. The images, offset by 50 arcsec in a + pattern, were corrected using geotran prior to mosaicking. The 2MASS point source catalog is overlaid on the field. Except for multiple sources mistakenly cataloged as a single 2MASS source, the positional fit is excellent to the edge of the field.

3.3.3 Larger Mosaics

Maps covering a larger area may be generated using the more generic IRAF tasks in the noao.imred.irred package. In particular, the tasks irmosaic, apphot.center and irmatch2d can be used in a manner analogous to the upsqiid tasks noted above. The only preconditions are that the input images have some spatial overlap with their neighbors

and that the data be taken in a regular N × M mosaic pattern. The process is a bit cumbersome in that one must manually generate and edit a file containing the relative coordinates of reference stars in adjacent images.

The irmosaic task combines the input images into a single N × M mosaic, in the same geometry that they were taken on the sky. It is important that one inspect the mosaic image to ensure that the component images are in the right order. The apphot.center task is then used to measure the centroids of pairs of stars which are common to adjacent images in the mosaic image. These coordinates are written to an output file, which must be manually edited so that only the X and Y coordinates of the stars remain. The irmatch2d task shifts the images in the original mosaic, using this coordinate file so that the adjacent stars match up, creating a single composite image.

NOTE: Because WHIRC is designed to exploit the good seeing at WIYN, particularly with WTTM, it is expected that it will be used mostly for deep mapping of relatively small areas. Other instruments, such as NEWFIRM, are better suited to wide-field surveys. The example given here utilizes a relatively small dither amplitude better suited to the xyget and nircombine tasks, but illustrates the workings of the larger irmosaic task.

3.3.3.1 Example Mosaic

The example given here is a 3 × 3 raster J band image of the V4334 Sgr field, with relatively small (30 arcsec) dithered offsets. In the X, Y image coordinate system, the mosaic starts at the lower left corner, moving up, to the right, down, to the right, and up in a boustrophedon pattern.

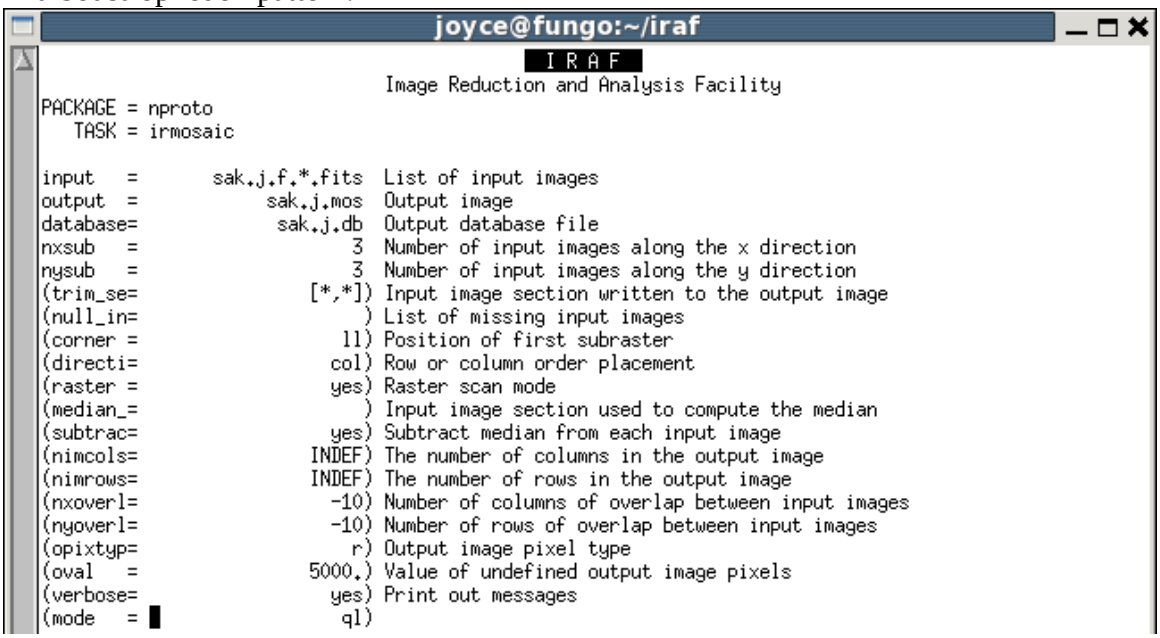
joyce@fungo:~/iraf
IRAF
Image Reduction and Analysis Facility
PACKAGE = nproto
TASK = irmosaic
input = sak.j.f.*.fits List of input images
output = sak.j.mos Output image
database= sak.j.db Output database file
nxsub = 3 Number of input images along the x direction
nysub = 3 Number of input images along the y direction
(trim_se= [*,*]) Input image section written to the output image
(null_in=) List of missing input images
(corner = ll) Position of first subraster
(directi= col) Row or column order placement
(raster = yes) Raster scan mode
(median_=) Input image section used to compute the median
(subtrac= yes) Subtract median from each input image
(nimcols= INDEF) The number of columns in the output image
(nimrows= INDEF) The number of rows in the output image
(nxoverl= -10) Number of columns of overlap between input images
(nyoverl= -10) Number of rows of overlap between input images
(opixtyp= r) Output image pixel type
(oval = 5000.) Value of undefined output image pixels
(verbose= yes) Print out messages
(mode = ql)

Figure 3.9: Parameter file for the IRAF task irmosaic.

This is reflected in the parameters ‘corner = ll’, ‘directi = col’, and ‘raster = yes’. The ‘nxoverl’ and ‘nyoverl’ parameters have been set to -10 with an intensity (‘oval’) of 5000

just to make the separation of the input images in the output mosaic obvious (Figure 3.10). It is easy to identify the common stars in the mosaic.

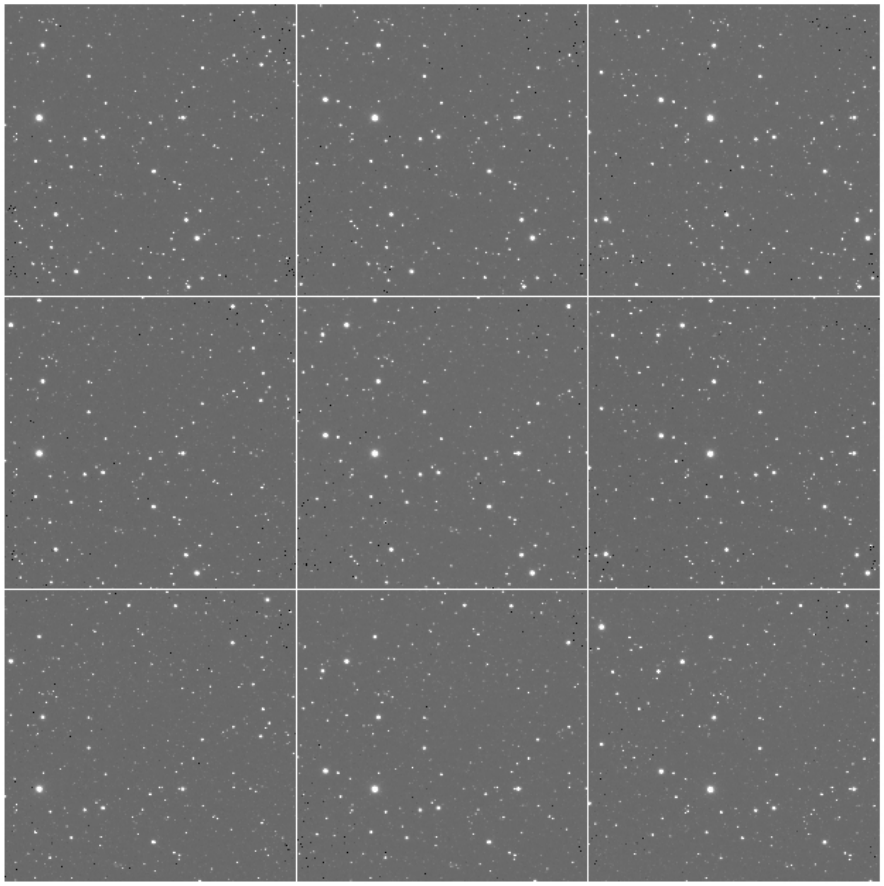

Figure 3.10: Output image sak.j.mos.fits generated by the irmosaic task.

One can now display the mosaic image and run the apphot.center task on it by centering the cursor on a star and hitting <spacebar> to measure the centroid. This must be done in pairs, measuring the same star in two adjacent subimages in the mosaic image. Even if the same star can be used in all images, it is important to maintain this order (e.g., ref star in image 1; same star in image 2; ref star in image 2; same star in image 3, etc…). You will end up with a text file of the sort '<mosaicimagename>.ctr.1'. Open up this file in an editor and delete all lines except those with X and Y coordinates, which should be in columns 1 and 2. Close the editor and write these two columns to a new file by the command fields <mosaicimagename>.ctr.1 1,2 > <mosaicimagename>.coords.

This coordinate file can then be used as the input to the task irmatch2d (Figure 3.11). The input image and database are those created by the irmosaic task, and the 'coords' file is the coordinate file generated by the apphot.center task. The 'alignme' must be set to 'coords' to use the coordinate file input. The result is shown in Figure 3.12.

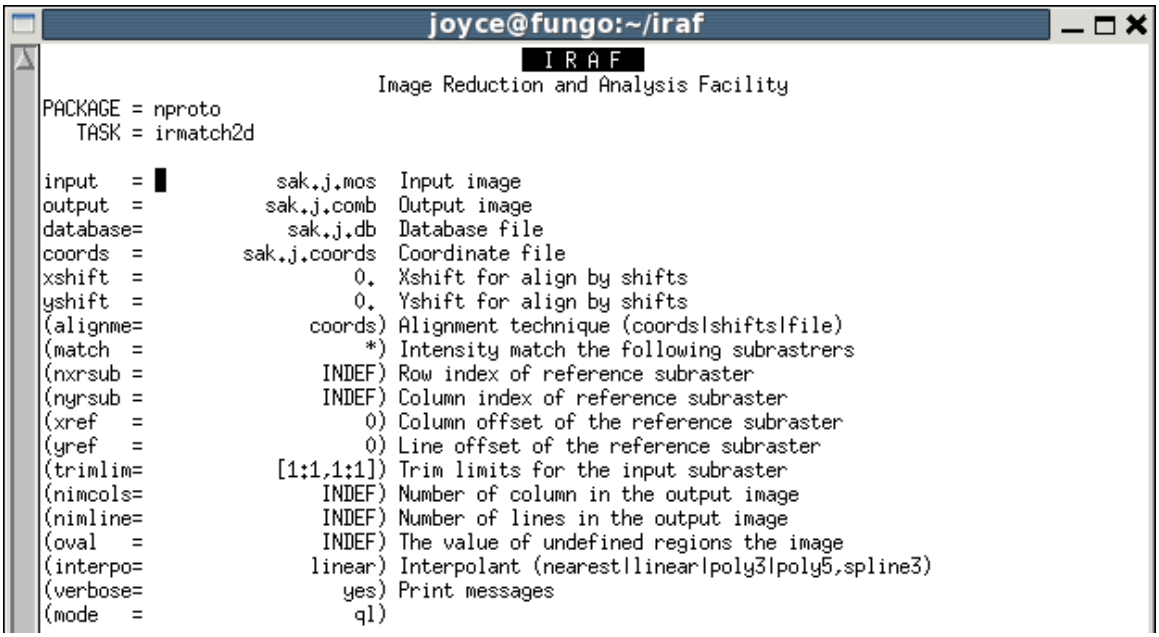
joyce@fungo:~/iraf
IRAF
Image Reduction and Analysis Facility
PACKAGE = nproto
TASK = irmatch2d
input = sak.j.mos Input image
output = sak.j.comb Output image
database= sak.j.db Database file
coords = sak.j.coords Coordinate file
xshift = 0. Xshift for align by shifts
yshift = 0. Yshift for align by shifts
(alignme= coords) Alignment technique (coords|shifts|file)
(match = *) Intensity match the following subrastrers
(nxrsub = INDEF) Row index of reference subraster
(nyrsub = INDEF) Column index of reference subraster
(xref = 0) Column offset of the reference subraster
(yref = 0) Line offset of the reference subraster
(trimlim= [1:1,1:1]) Trim limits for the input subraster
(nimcols= INDEF) Number of column in the output image
(nimline= INDEF) Number of lines in the output image
(oval = INDEF) The value of undefined regions the image
(interpo= linear) Interpolant (nearest|linear|poly3|poly5,spline3)
(verbose= yes) Print messages
(mode = ql)

Figure 3.11: Parameter file for the IRAF task irmatch2d.

Figure 3.12: Aligned image resulting from the irmatch2d task.

4.0 Appendices

4.1 Appendix A – Observational Test of Pupil Ghost Removal

On the (apparently) photometric night of 9 July 2009 UT we tested the pupil ghost removal technique described in section 2.4.2 by sampling a region approximately 80 arcsec square centered on the array. An arbitrary star (Ks ~ 11) on the outskirts of M13 was measured using four 5 × 5 dither sequences of amplitude 10 arcsec, each of which was offset by 18 arcsec in both RA and DEC from the center of the array in the four possible directions. This gave high spatial sampling of the area around the center of the pupil ghost (approximately [1032:1002] on the array). The experiment was done in the Ks filter where the pupil ghost is most pronounced. The focus was checked between each of the four sequences to ensure that the image quality did not drift significantly during the experiment; seeing was modest (0.6 – 0.7 arcsec).

Data reduction was done in the standard way. The 100 frames were median combined to generate a sky frame, which was subtracted from each of the input frames to yield the sky-subtracted images.

Good quality dome flats in the J and Ks filters were obtained using the standard "lights on, lights off" technique, subtracted and then corrected for bad pixels using the fixpix task and the WHIRC bad pixel mask. The flats were normalized using regions [400:600,400:1600] and [1400:1600,400:1600], which avoids the pupil ghost area, and any other pixels of value < 0.02 were set to 1.0 to avoid arithmetic blowup in the flattened images. These flats, which contain the pupil ghost, are referred to as the uncorrected flats.

The template for removal of the pupil ghost was generated as follows:

- The normalized Ks and J band flats were ratioed to give a flat ratio image
- A new 2048 × 2048 image with pixel value = 1.0 was generated with mkimage
- The central 800 × 800 of the flat ratio image centered on the pupil ghost [625:1424,601:1400] was copied into the new image
- The image was smoothed using the gauss task with sigma = 31
- Subtract 1.0 from the result to yield the pupil ghost template image

The uncorrected flats were run through the rmpupil task using the pupil ghost template image as the 'pupil', and the pupil mask file pupmask.pl as the 'pupilmask' to restrict the procedure to the central 600 pixels [725:1324,725:1324]. The parameter 'outtype' was set to "sdiff", since one is subtracting the ghost, and the column and line smoothing were set to 20. The resulting flats were the "nopupil" flats.

The sky subtracted images were flattened with both the uncorrected and nopupil flats, yielding three datasets of 100 images each:

- Sky-subtracted, unflattened images
- Uncorrected flattened images
- Pupil corrected flattened images

The selected star was measured using apphot with a fairly large (4 arcsec diameter) aperture to minimize the effects of seeing or focus drifts for all three datasets. The X and Y coordinates were translated to radial distance from the pupil ghost center at [1032:1002]; this was used as the abscissa for plotting the measured signal levels (Fig. 4.1).

The figure illustrates that the uncorrected flat does have the anticipated effect of reducing the point to point scatter in the measured flux, but introduces a very significant (0.15 mag) systematic artifact as a result of the pupil ghost. The additional illumination in the ghost results in signal levels which are too faint. The bottom panel of the figure demonstrates that the pupil removal procedure has effectively eliminated this artifact.

The rms scatter in the pupil-corrected flattened images (0.024 mag) is significantly less than that in the unflattened images (0.035 mag), but is still larger than earlier tests with a 9 × 9 raster of a standard star field had yielded. The current experiment of measuring 100 positions over a 40 minute time frame does make assumptions about photometric stability over this time frame during a time of year not known for clear weather. A plot of the data with time (Fig. 4.2) does appear to show some systematic variation, as well as a slope which is probably a result of atmospheric extinction ($\Delta z \sim 0.2$) during the sequence. The p-p scatter around a low-order fit is ~ 0.05 – 0.06 mag, equivalent to an rms ~ 0.015 mag.

An alternative approach is to image a field which is sufficiently crowded to spatially sample the area within and around the pupil ghost, as this would not be sensitive to changes in sky transparency. Unfortunately, real-life stellar fields of sufficient density are usually confusion limited with many fainter sources contributing a background which calls into question the photometric reliability of relatively coarse resolution surveys such as 2MASS.

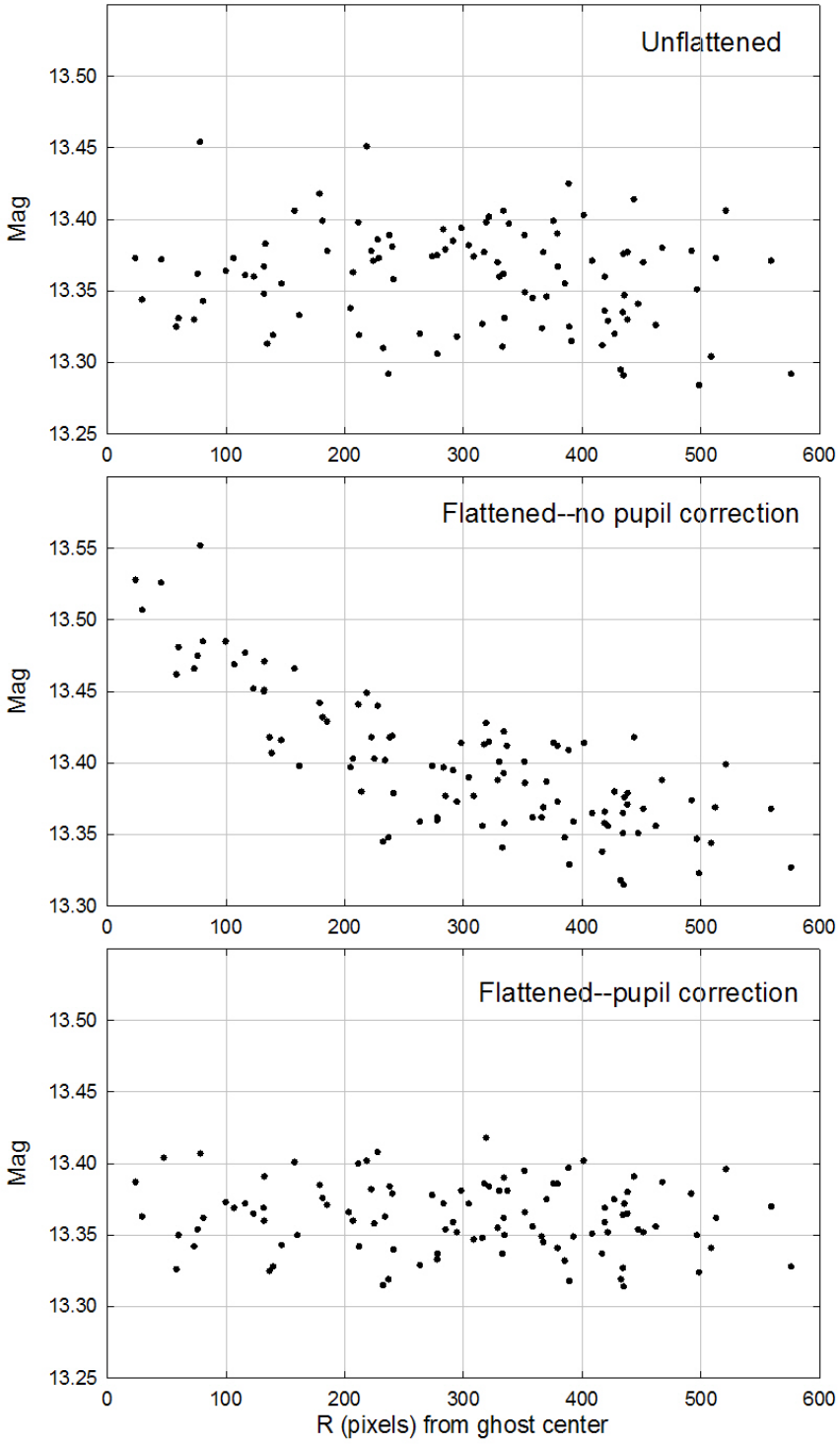

Figure 4.1: Plots of the magnitude of a Ks ~ 11 star measured in a 4 arcsec aperture using the IRAF task apphot over a 10 × 10 raster centered on the WHIRC pupil ghost. The magnitude scale is not calibrated to a photometric standard. The same dataset is plotted for the sky-subtracted, unflattened data (top panel), data flattened with no correction for the pupil ghost (middle panel), and data flattened with correction for the pupil ghost (bottom panel).

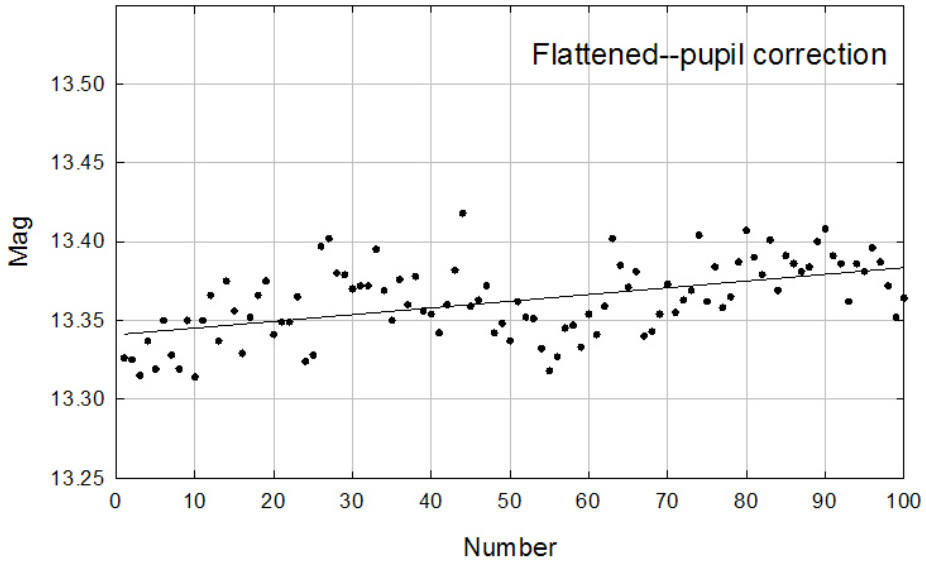

Figure 4.2: The pupil-corrected flattened data from the lower panel of Fig. 4.1, but plotted as a function of time, in the order the data were obtained. The line is a linear fit to the data and probably represents atmospheric extinction, since the airmass increased over the span of the observations by approximately 0.20.